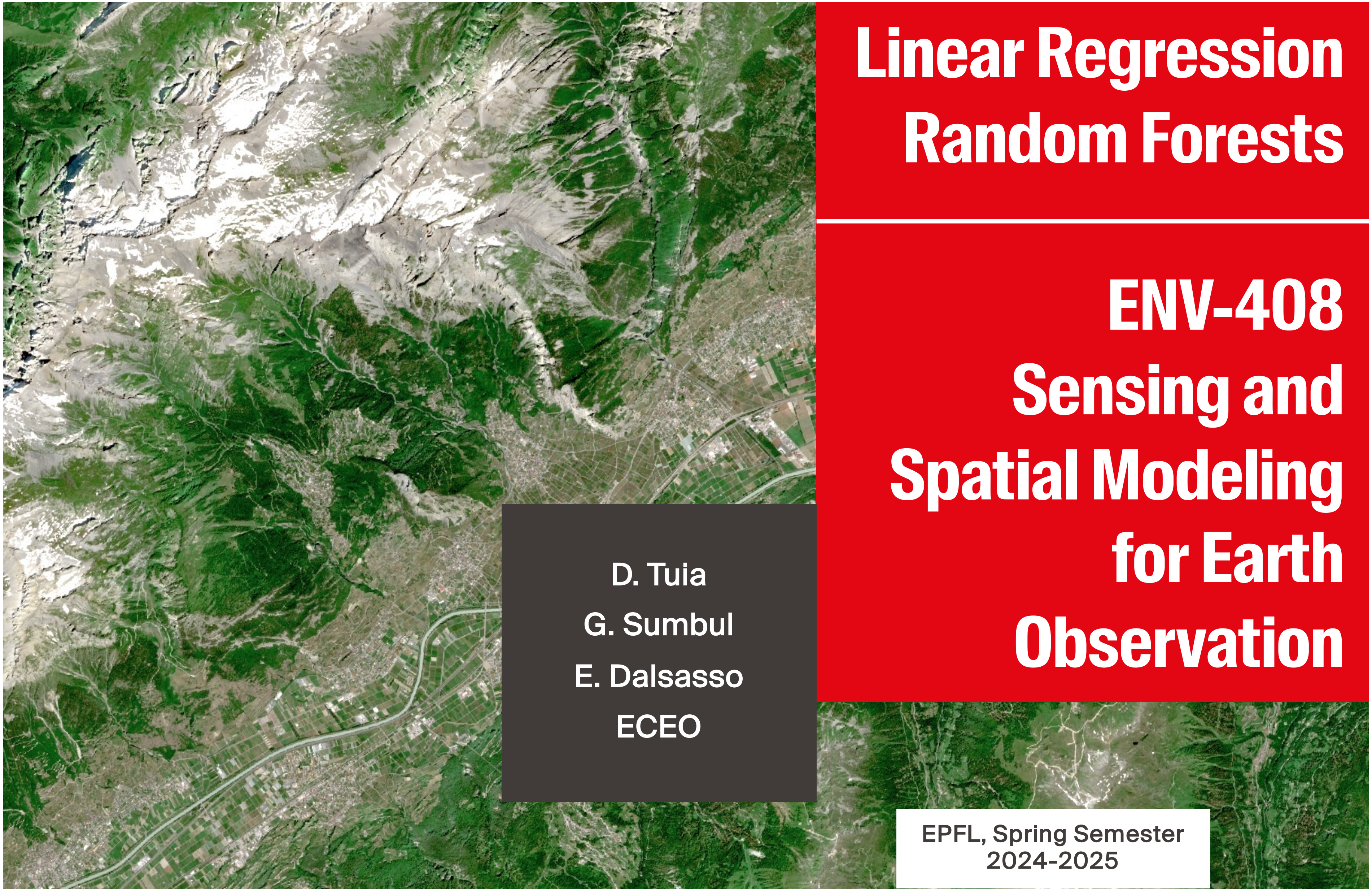


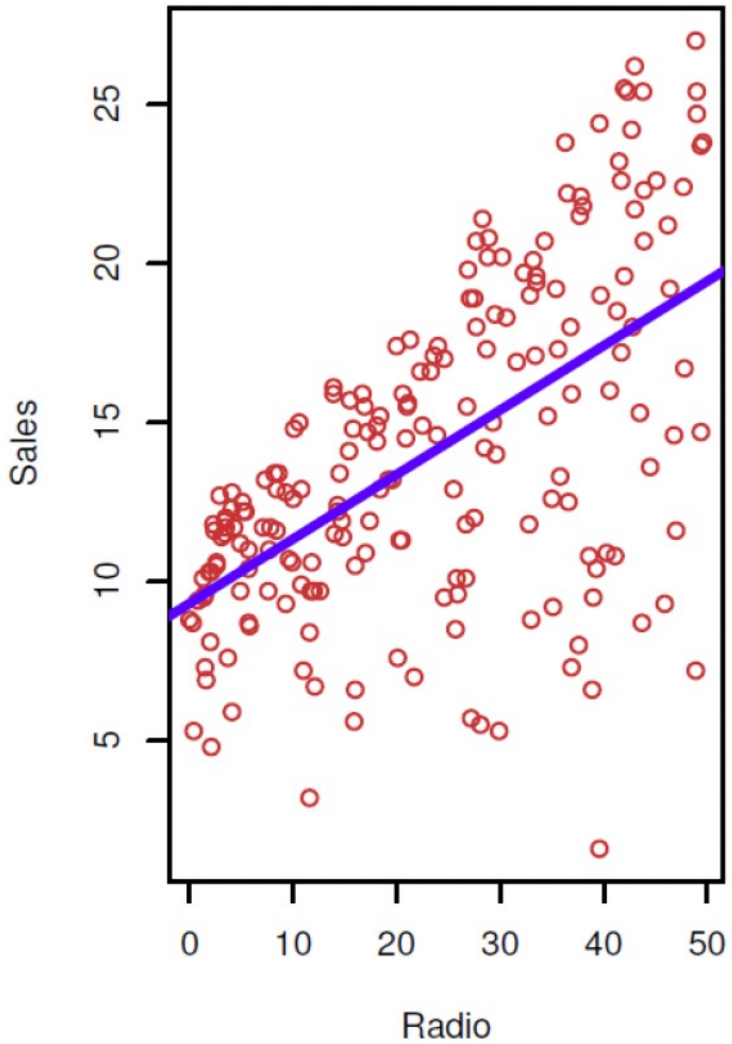
Linear Regression
Random Forests

ENV-408
Sensing and Spatial Modeling for Earth Observation

D. Tuia
G. Sumbul
E. Dalsasso
ECEO

EPFL, Spring Semester 2024-2025

EPFL
École polytechnique fédérale de Lausanne



Outline

Linear Regression Fundamentals

Multivariate Linear Regression

Model Evaluation

Decision Trees

Bagging

Random Forests

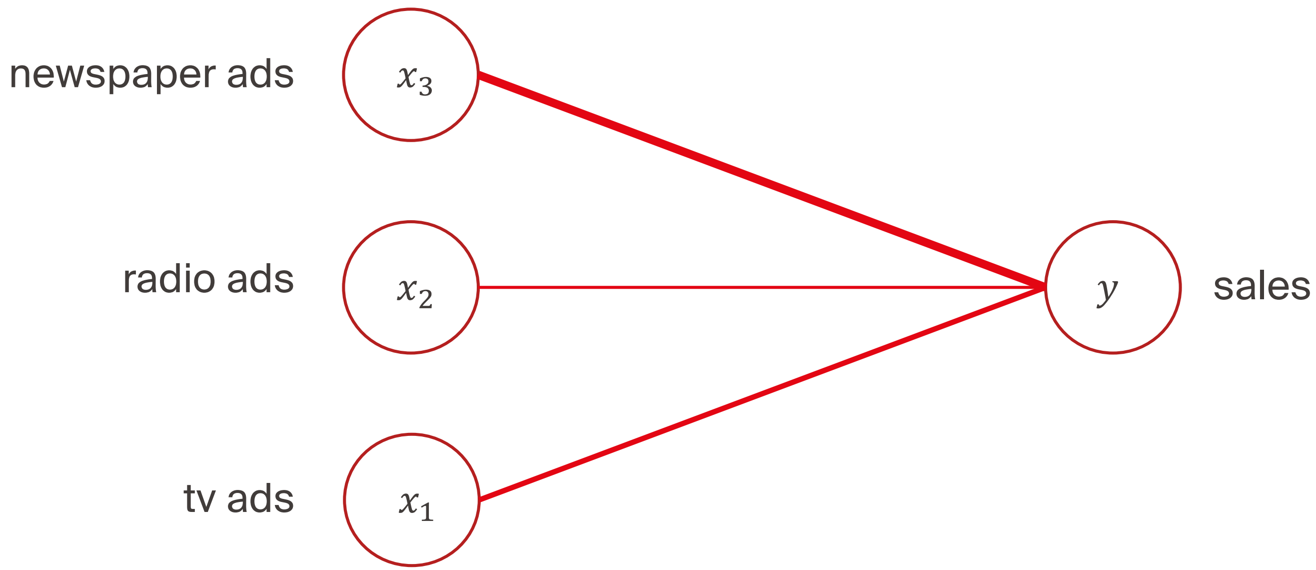
Linear regression

- An approach for supervised learning:
  - the most used model in science → has been around for a long time
  - simple and useful statistical learning method to predict quantitative responses → ideal for many real-world problems
  - forms a basis for many complex methods → learning it helps to understand complex methods

1. What is linear regression?
2. How to ‘fit’ a linear regression model?
3. How to evaluate a linear regression model?
4. How to select features for linear regression?

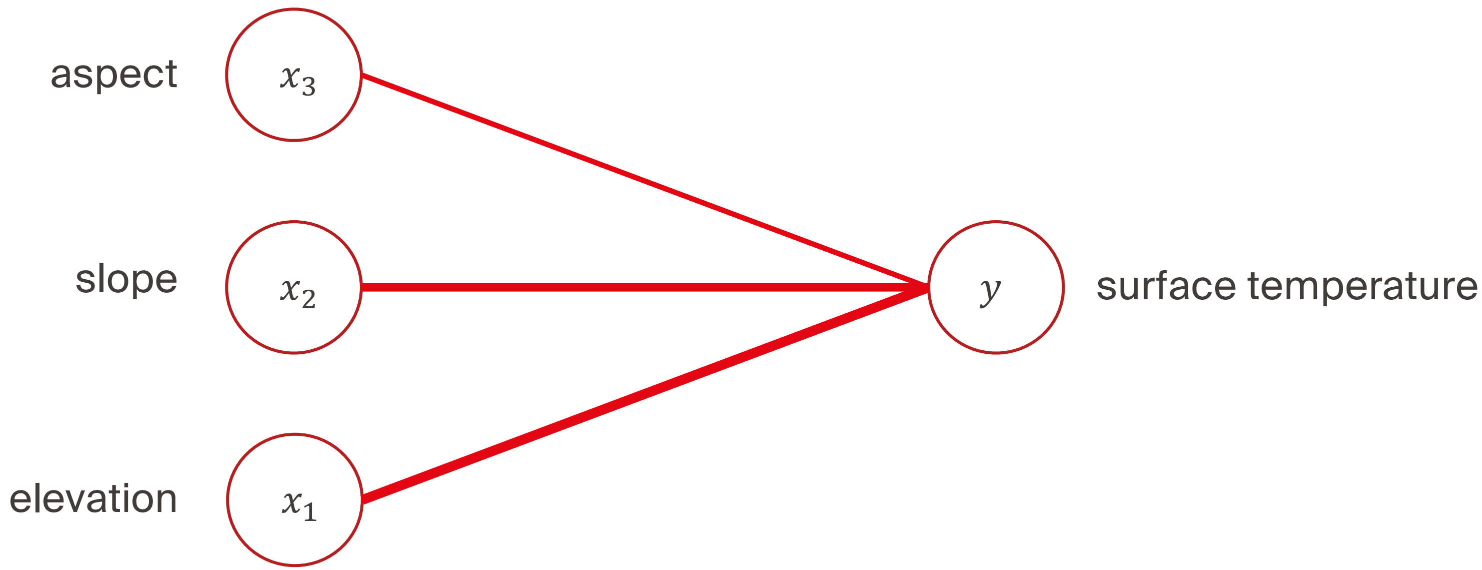
What is linear regression?

- Measuring relationships between variables



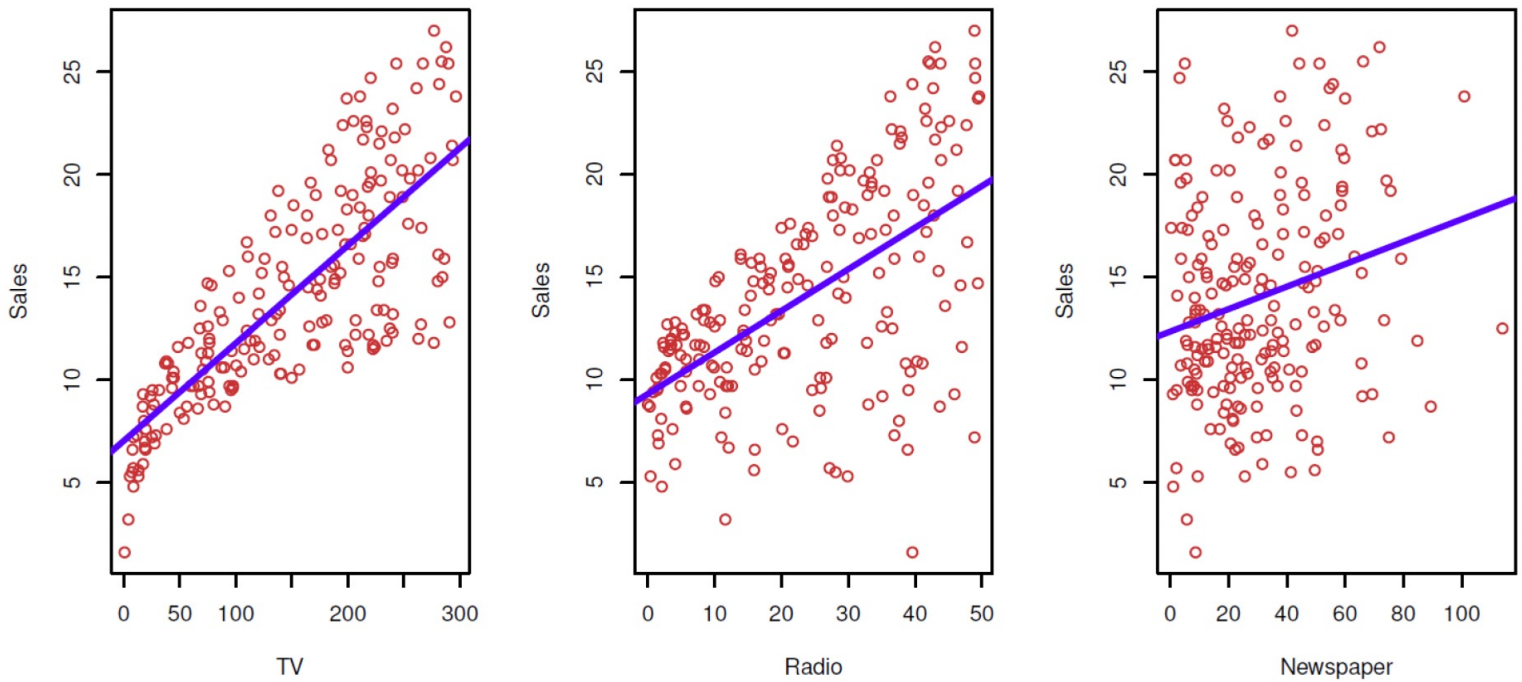
What is linear regression?

- Measuring relationships between variables



What is linear regression?

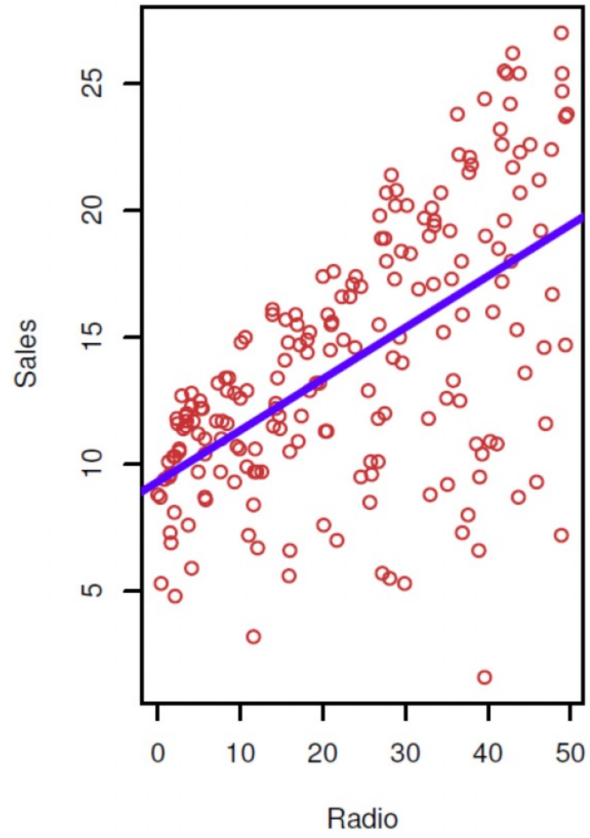
- Measuring relationships between variables through;
  - data $D = \left\{\left(\left(x_1^i, x_2^i, x_3^i\right), y^i\right)\right\}_{i=1}^{N}$



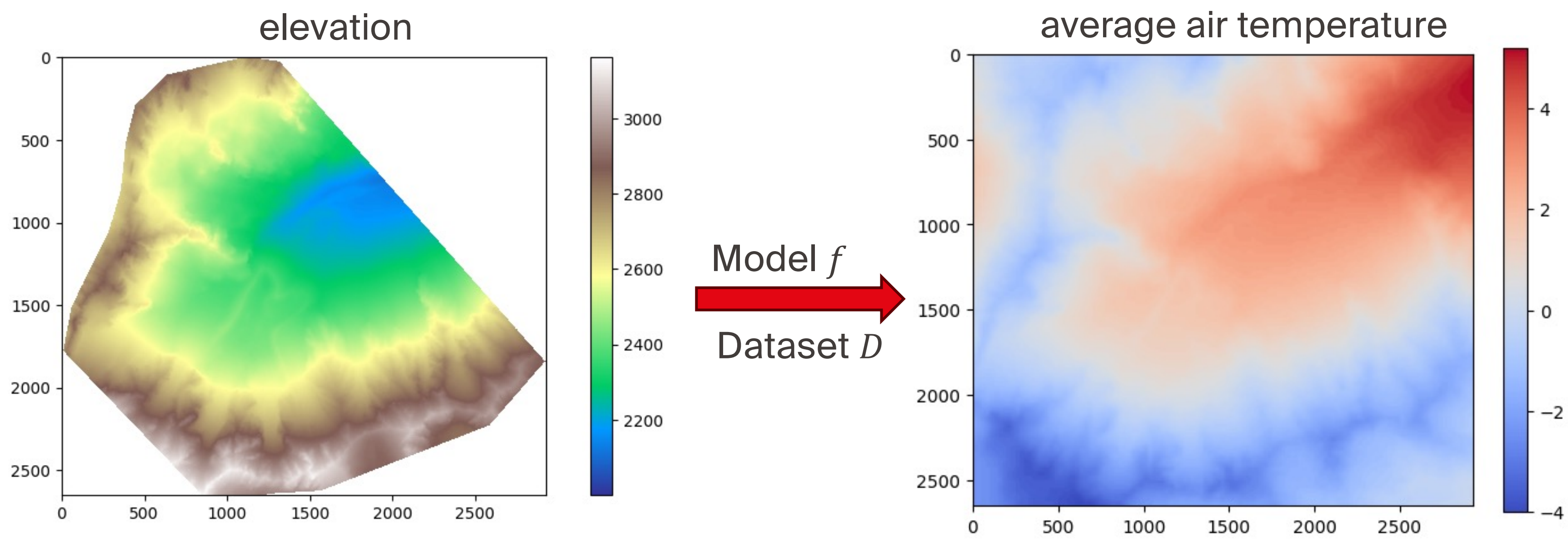
What is linear regression?

- Measuring relationships between variables through data for answering questions about the relationship of variables

  - Is there a relationship between advertising budget and sales?
  - How strong is the relationship?
  - Is there a synergy among the advertisement media?



How to 'fit' a linear regression model?

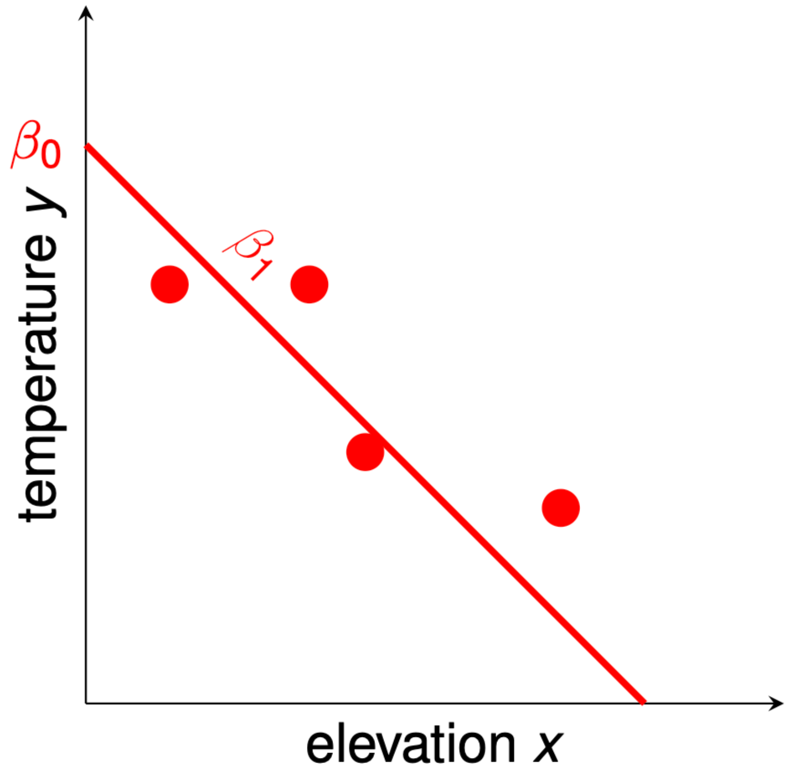


- A linear regression model is defined by the y-intercept β_0 and the slope β_1 as:

$$\hat{y} = f(x; \boldsymbol{\beta}) = \beta_0 + x\beta_1$$

which maps each x (e.g., an elevation) to the estimated continuous value $\hat{y}$ (e.g., the average surface temperature prediction).

How to 'fit' a linear regression model?

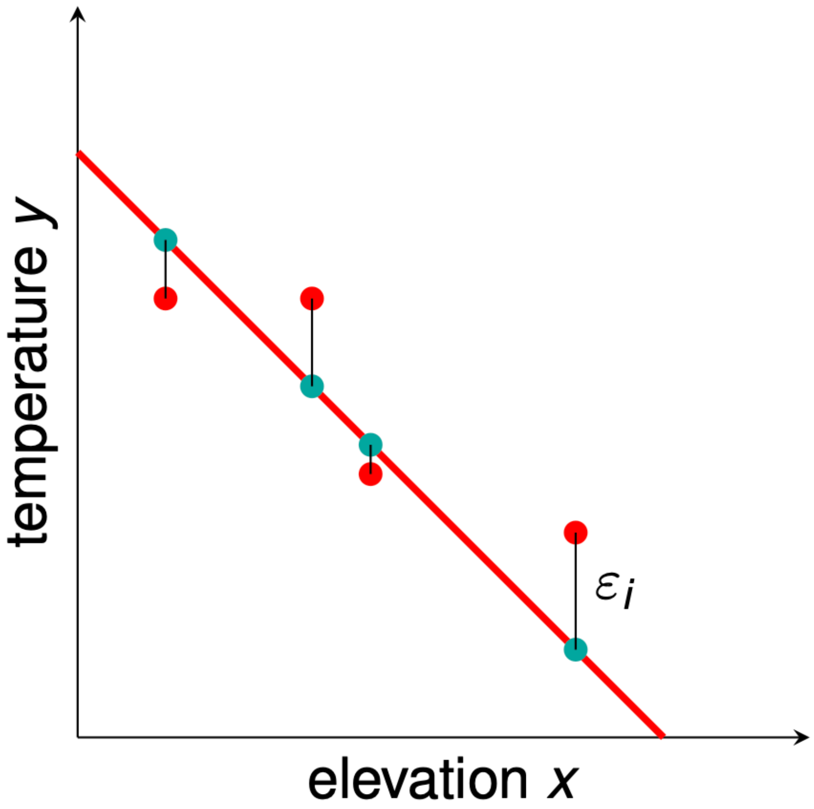


- Assume that we have a dataset of N samples $D = \{(x^i, y^i)\}_{i=1}^{N}$

- How to find good parameters $\widehat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1)$ for a linear regression model $\hat{y} = f(x; \beta) = \beta_0 + x\beta_1$?

→ How to 'fit' a linear regression model?

How to 'fit' a linear regression model?



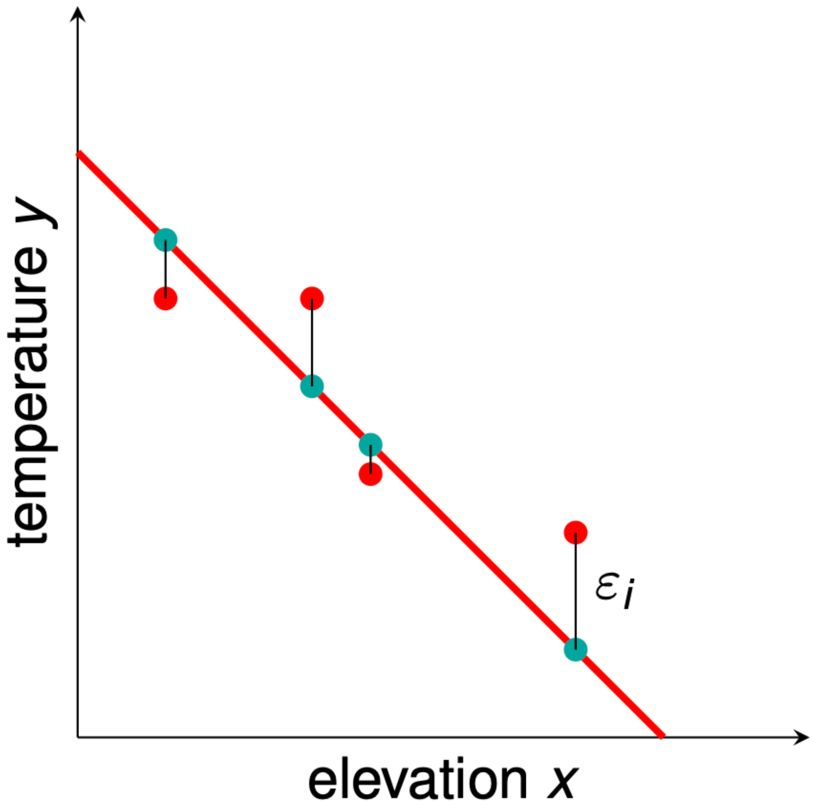
- For a selection of (β_0, β_1), the model f estimates $\hat{y}^i$ for each sample x^i with error ε^i (residual):

$$\hat{y}^i = f(x^i; \beta) = \beta_0 + x^i\beta_1 = y^i \pm \varepsilon^i$$

- The best parameters $\widehat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1)$ minimize the residuals over the dataset.

→ How to 'fit' a linear regression model?
  - By minimizing the residuals

How to 'fit' a linear regression model?

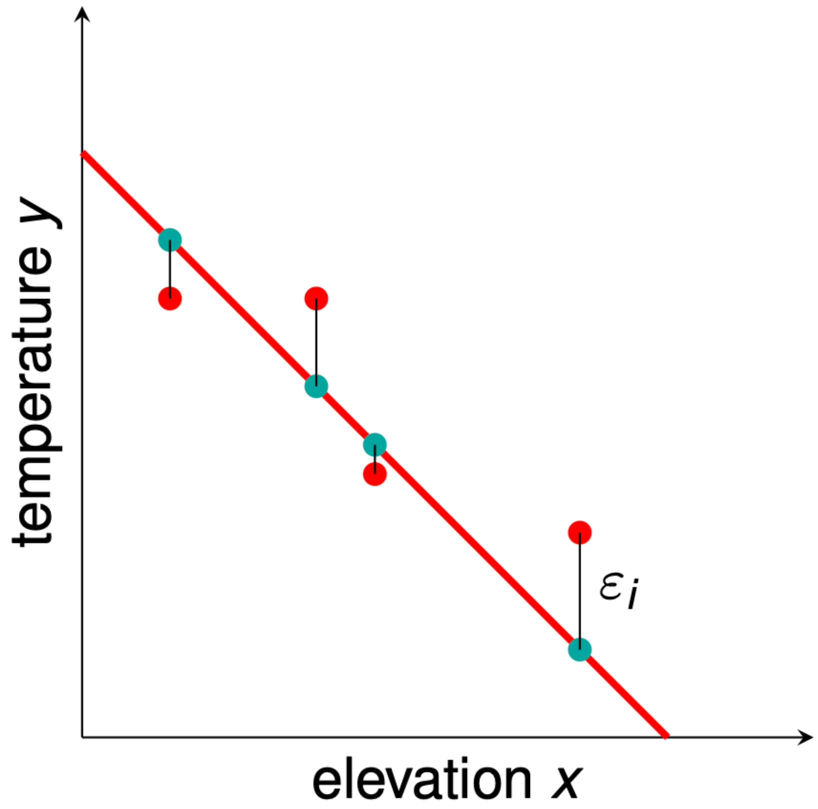


- How to minimize residuals?
  - Mean absolute error (MAE) over the estimations and dataset

$$\frac{1}{N}\sum_{i=1}^{N} |\hat{y}^i - y^i|$$

  - Very intuitive, but does not punish big errors

How to 'fit' a linear regression model?



- How to minimize residuals?
  - Mean squared error (MSE) over the estimations and dataset

$$\frac{1}{N}\sum_{i=1}^{N}\left(\hat{y}^i - y^i\right)^2$$

  - does punish big errors by a square penalty

How to 'fit' a linear regression model?

- The best parameters $\widehat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1)$ are found by minimizing MSE over dataset through ordinary least squares (OLS) solution.

$$\hat{\beta}_0, \hat{\beta}_1 = \underset{\beta_0,\beta_1}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} \left(f\left(x^i; \boldsymbol{\beta}\right) - y^i\right)^2 = \underset{\beta_0,\beta_1}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} \left(\beta_0 + x^i \beta_1 - y^i\right)^2$$

- Solving a minimization problem:
  1. Calculate partial derivatives $\frac{\partial \mathrm{MSE}}{\partial \beta_0}$ and $\frac{\partial \mathrm{MSE}}{\partial \beta_1}$
  2. Set derivatives to zero: $\frac{\partial \mathrm{MSE}}{\partial \beta_0} = 0, \frac{\partial \mathrm{MSE}}{\partial \beta_1} = 0$
  3. Solve for $\hat{\beta}_0$ and $\hat{\beta}_1$

How to 'fit' a linear regression model?

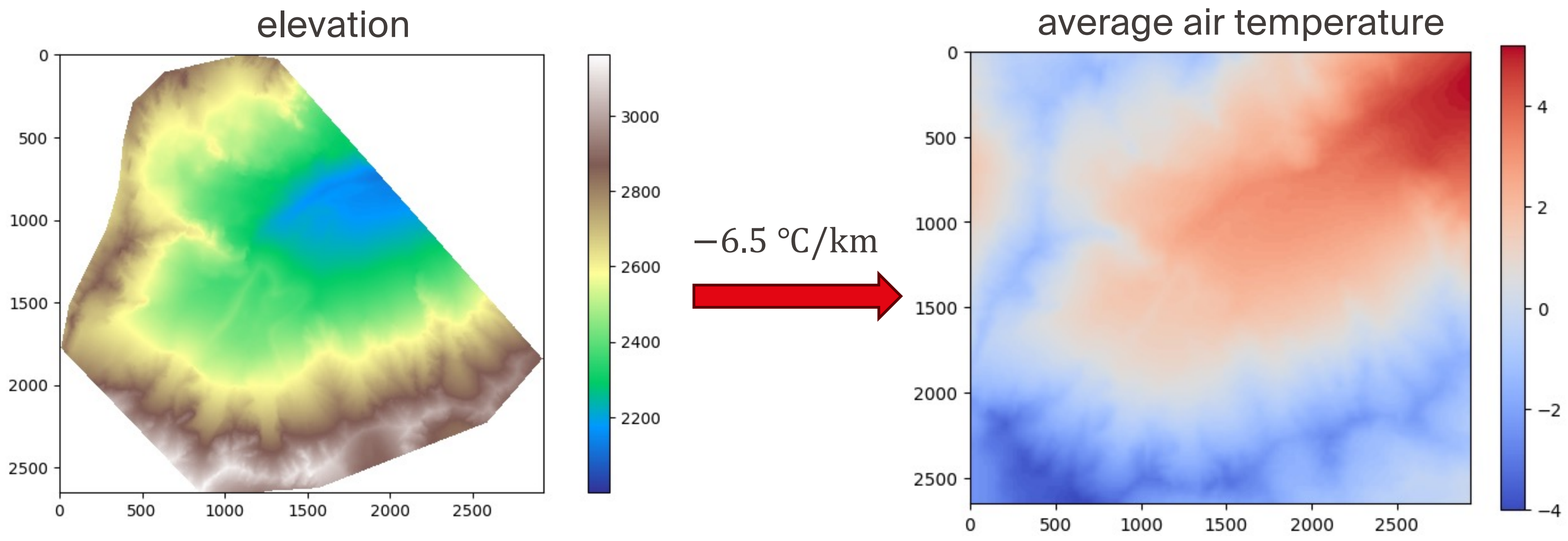
- The best parameters $\widehat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1)$ are found by minimizing MSE over dataset through ordinary least squares (OLS) solution.
  - By solving the minimization problem, the closed form OLS solution is obtained as:

$$\hat{\beta}_0 = \frac{1}{N}\sum_{i=1}^{N} y^i - \beta_1 \frac{1}{N}\sum_{i=1}^{N} x^i = \bar{y} - \beta_1 \bar{x}$$

$$\hat{\beta}_1 = \frac{\frac{1}{N}\sum_i (x^i - \bar{x})(y^i - \bar{y})}{\frac{1}{N}\sum_i (x^i - \bar{x})^2} = \frac{\mathrm{Cov}(x, y)}{\mathrm{Var}(x)}$$

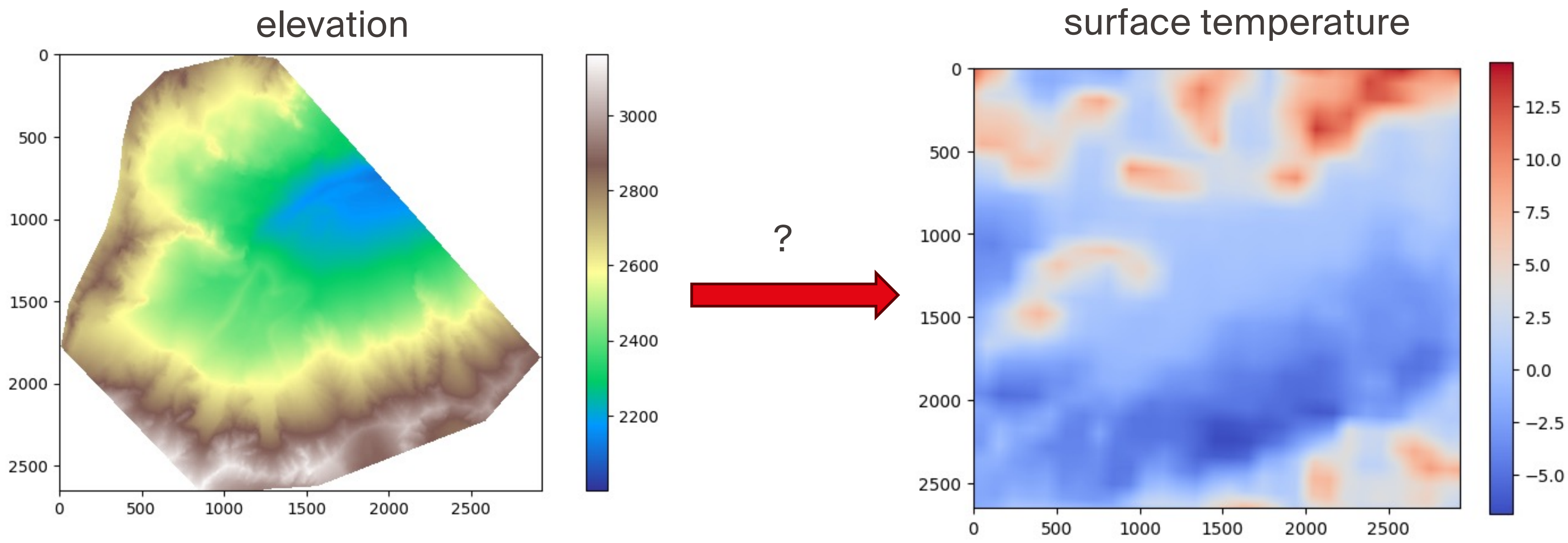
How to 'fit' a linear regression model?

- Some linear relations can be well-described by a single variable $\widehat{\boldsymbol{\beta}}$.
  - Temperature generally decreases with increasing elevation (known as environmental lapse rate).



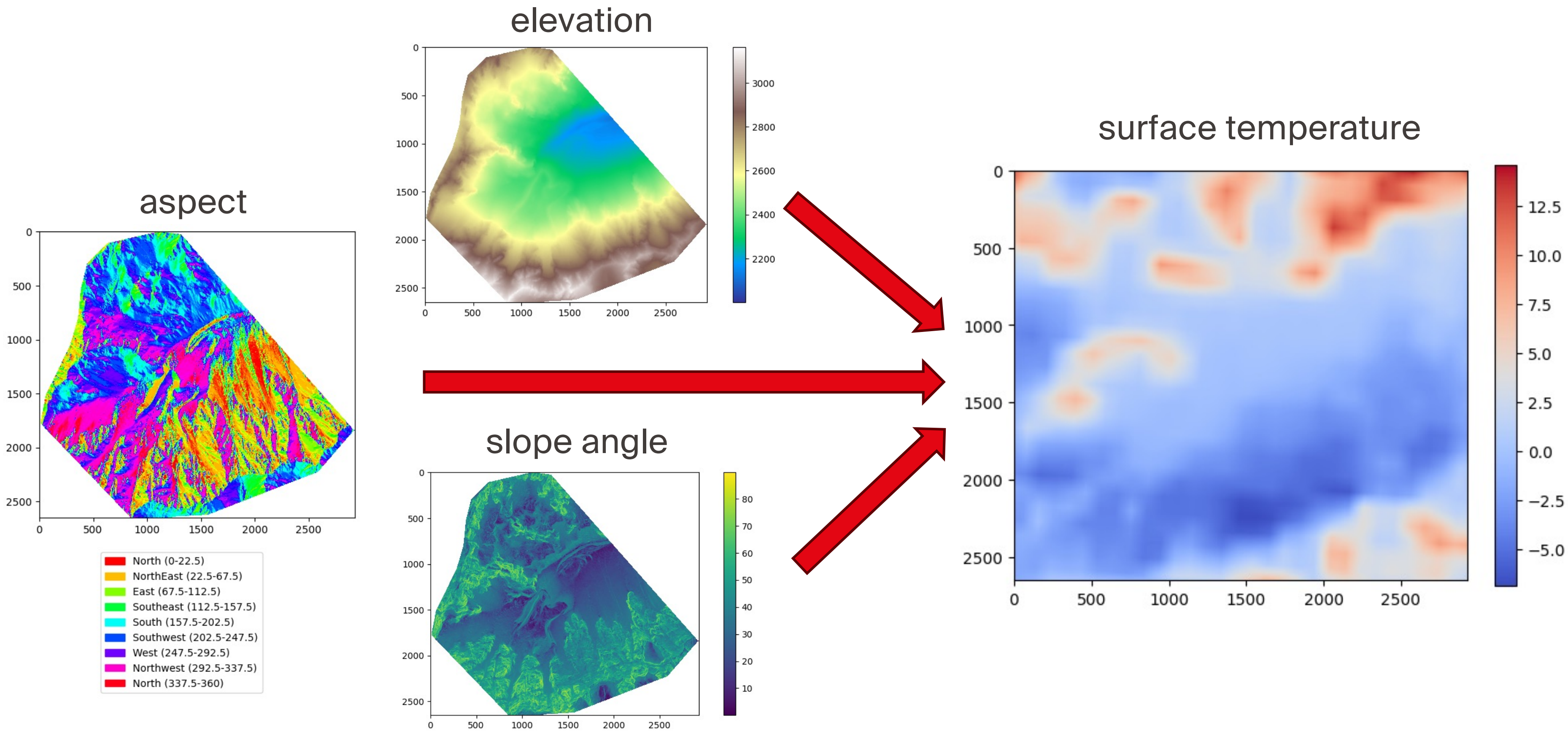
How to 'fit' a linear regression model?

- Most relationships are non-linear, and thus can't be described by a linear model with a single variable $\hat{\beta}$. → univariate linear regression



How to 'fit' a linear regression model?

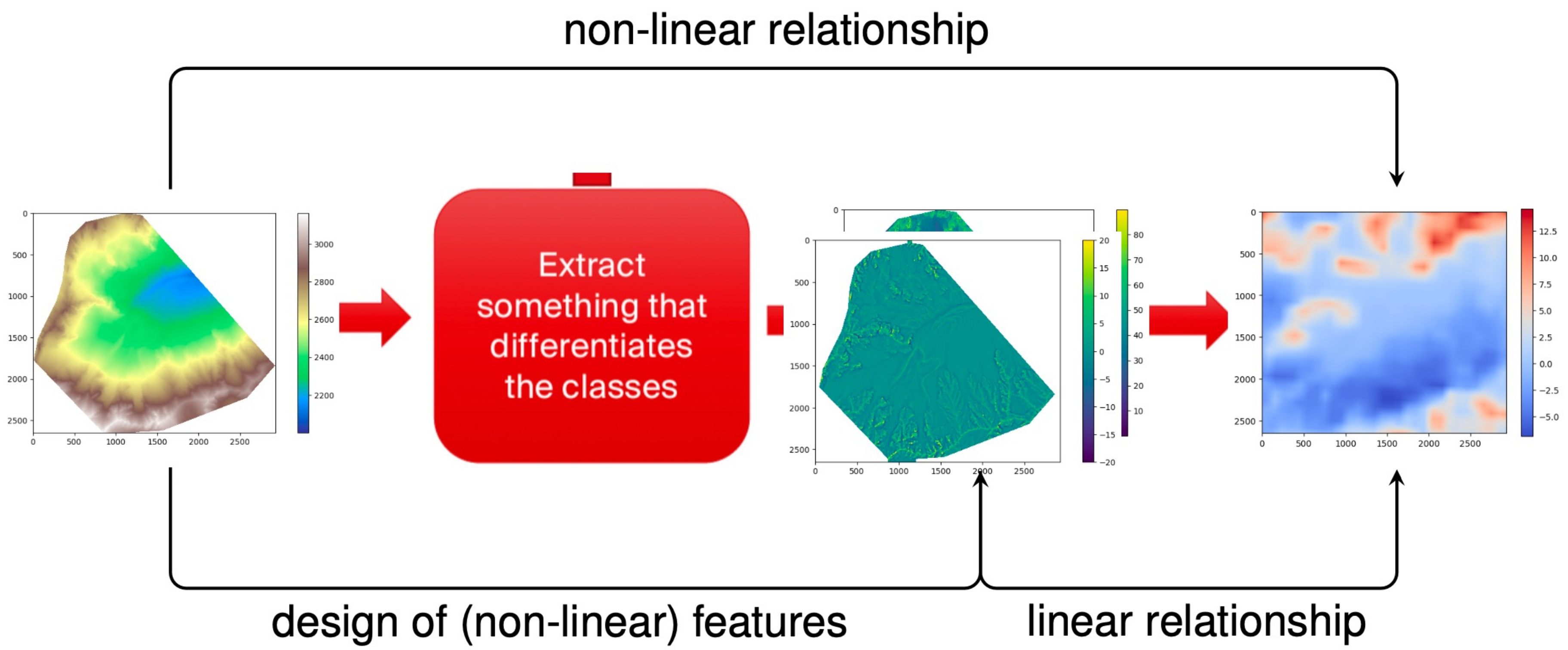
- Some non-linear relationships can be linear when combined together with good feature design. → multivariate linear regression



Tuia / Sumbul / Dalsasso

How to 'fit' a linear regression model?

- Some non-linear relationships can be linear when combined together with good feature design. → multivariate linear regression



EPFL

Multivariate Linear Regression

Tuia / Sumbul / Dalsasso

Multivariate linear regression

- Univariate linear regression model $\hat{y} = f(x; \beta_0, \beta_1)$ is formulated for scalar x and y with two parameters β_0, β_1.

$$\hat{y}^i = \beta_0 + \underbrace{x^i}_{\text{elevation}}\beta_1$$

- Multivariate linear regression model is formulated for:
  - input sample $\mathbf{x}^i = (x_1^i, x_2^i, \dots, x_p^i)$ of p scalar features
  - corresponding scalar target y^i
  - y-intercept β_0 and one parameter for each feature $\boldsymbol{\beta} = (\beta_0, \beta_1, \beta_2, \dots, \beta_p)$

$$\hat{y}^i = \beta_0 + \underbrace{x_1^i}_{\text{elevation}}\beta_1 + \underbrace{x_2^i}_{\text{slope}}\beta_2 + \underbrace{x_3^i}_{\text{aspect}}\beta_3 + \cdots + x_p^i\beta_p = \boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}^i$$

Multivariate linear regression

- All the samples of a multivariate dataset $D = \{(\boldsymbol{x}^i, y^i)\}_{i=1}^{N}$ can be expressed in vectors with N data samples and p features.

$$\mathbf{X} = \begin{pmatrix} 1 & x_1^1 & \cdots & x_p^1 \\ 1 & x_1^2 & & x_p^2 \\ & \vdots & \ddots & \vdots \\ 1 & x_1^N & \cdots & x_p^N \end{pmatrix}, \mathbf{Y} = \begin{pmatrix} y^1 \\ y^2 \\ \vdots \\ y^N \end{pmatrix}$$

- Then, for all the samples, the multivariate linear regression model with $p + 1$ parameters $\boldsymbol{\beta} = (\beta_0, \beta_1, \dots, \beta_p)$ is written as:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

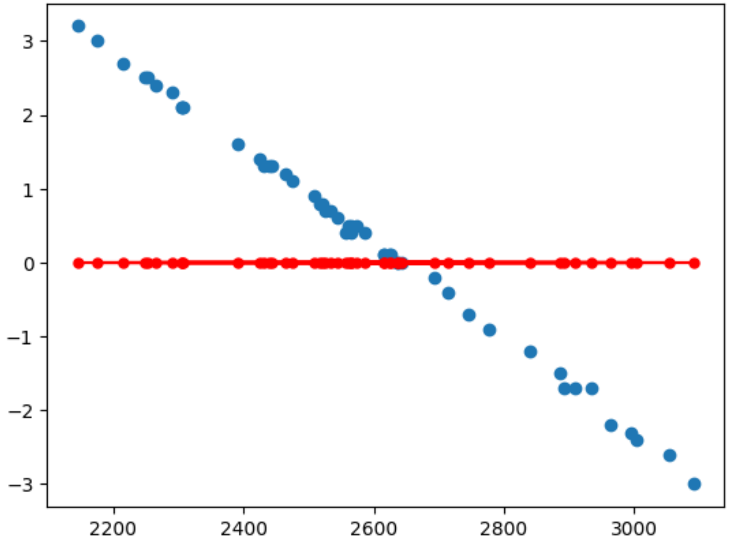
Multivariate linear regression

- The best parameters $\widehat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_p)$ are found by minimizing MSE over dataset through OLS solution.

$$\widehat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \|\mathbf{X}\boldsymbol{\beta} - \mathbf{Y}\|^2 = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} (\mathbf{Y}^{\mathrm{T}}\mathbf{Y} - 2\boldsymbol{\beta}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{Y} + \boldsymbol{\beta}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}\boldsymbol{\beta})$$

- Solving a minimization problem:
  1. Calculate partial derivative $\frac{\partial \mathrm{MSE}}{\partial \boldsymbol{\beta}} = -2\mathbf{X}^{\mathrm{T}}\mathbf{Y} + \mathbf{2}\mathbf{X}^{\mathrm{T}}\mathbf{X}\boldsymbol{\beta}$
  2. Set derivatives to zero: $\frac{\partial \mathrm{MSE}}{\partial \boldsymbol{\beta}} = -2\mathbf{X}^{\mathrm{T}}\mathbf{Y} + \mathbf{2}\mathbf{X}^{\mathrm{T}}\mathbf{X}\boldsymbol{\beta} = \mathbf{0}$
  3. Solve for $\widehat{\boldsymbol{\beta}} = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{Y}$

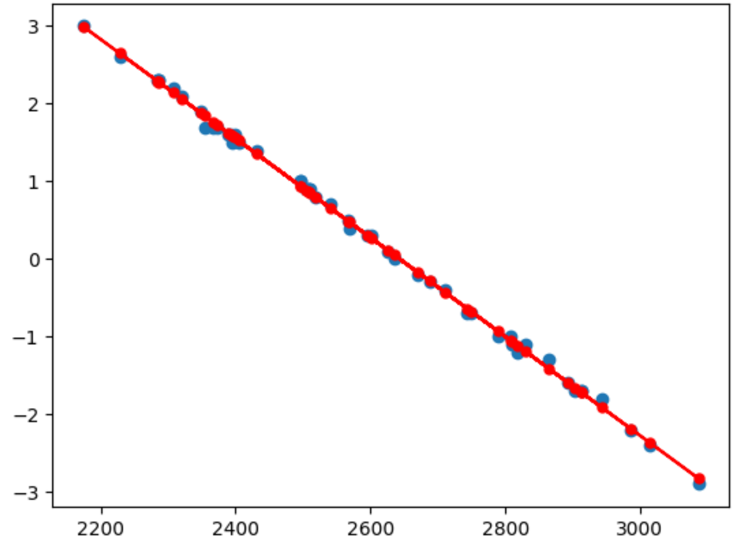
Multivariate linear regression



$\boldsymbol{\beta} = (0,0,0,0, \dots)$

$$\widehat{\boldsymbol{\beta}} = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{Y}$$

single line of code!



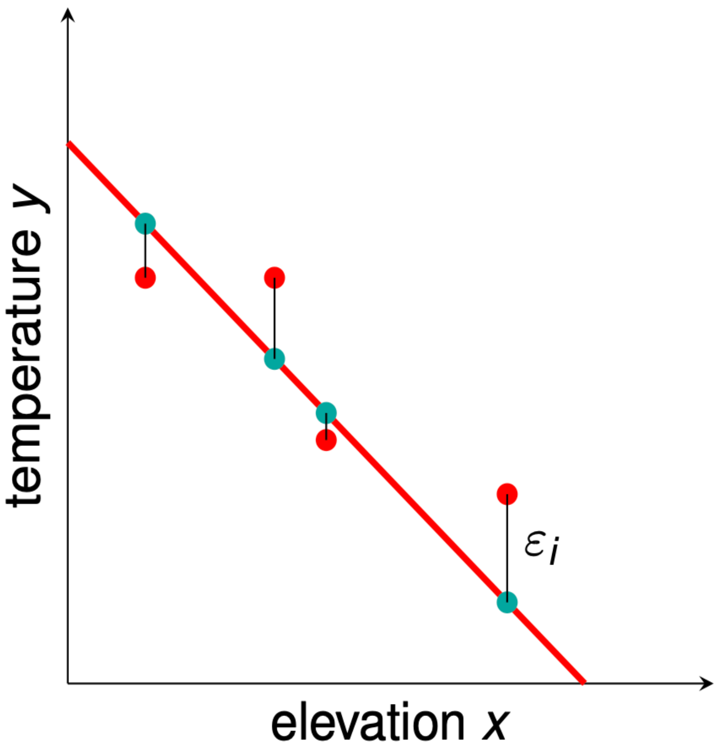
$\boldsymbol{\beta} = (-0.1, 0.2, 0.5, -0.5, \dots)$

EPFL

Linear Regression Model Evaluation

EPFL

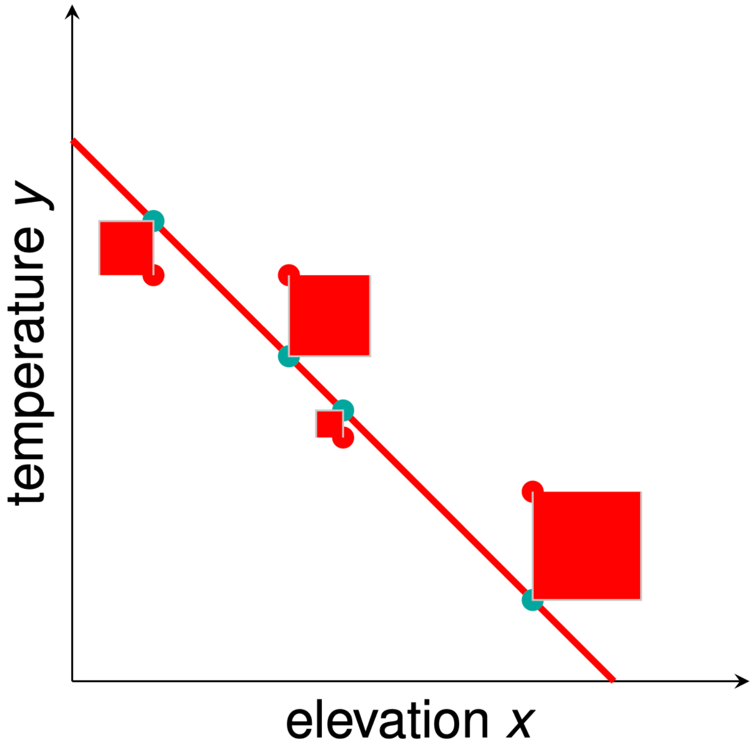
How to evaluate a linear regression model?



- After fitting the model with $\hat{\boldsymbol{\beta}}$, did we capture the underlying relationship between x and y?

1. How large is the error $\boldsymbol{\varepsilon}$?
   - Residual sum of squares (RSS)
   - Residual standard error (RSE)
   - R^2 metric
2. How close is the estimated $\hat{\boldsymbol{\beta}}$ to the true $\boldsymbol{\beta}$?
   - Standard errors (SE) for each parameter
3. How the relationship is significant according to dataset size?
   - Student's t-test

How to evaluate a linear regression model?

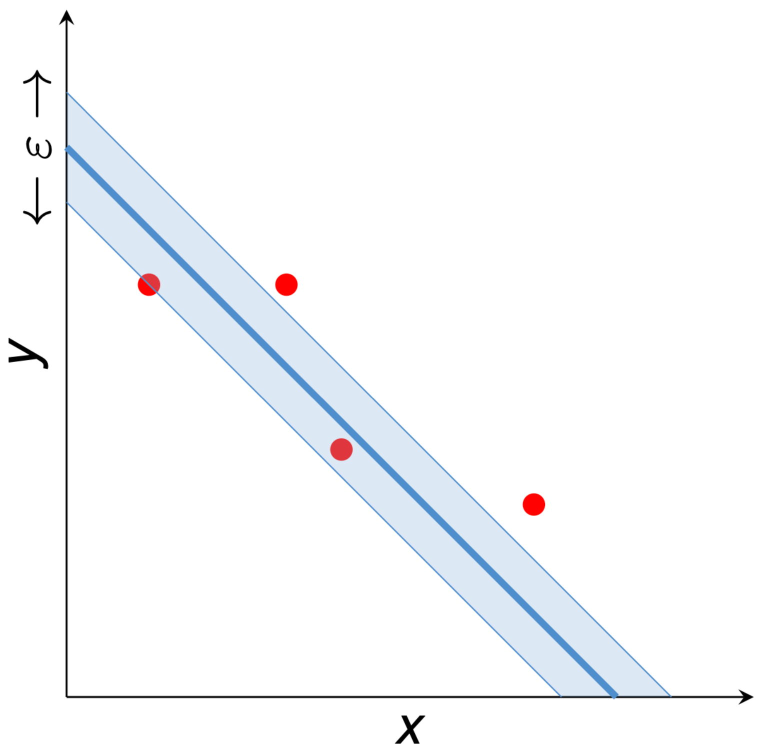


- Residual sum of squares (RSS) is defined as:

$$\mathrm{RSS} = \sum_{i=1}^{N} \left(y^i - \hat{y}^i\right)^2$$

- It increases with the dataset size N: hard to evaluate a model independent from N.

How to evaluate a linear regression model?



- Residual standard error (RSE) is defined as:

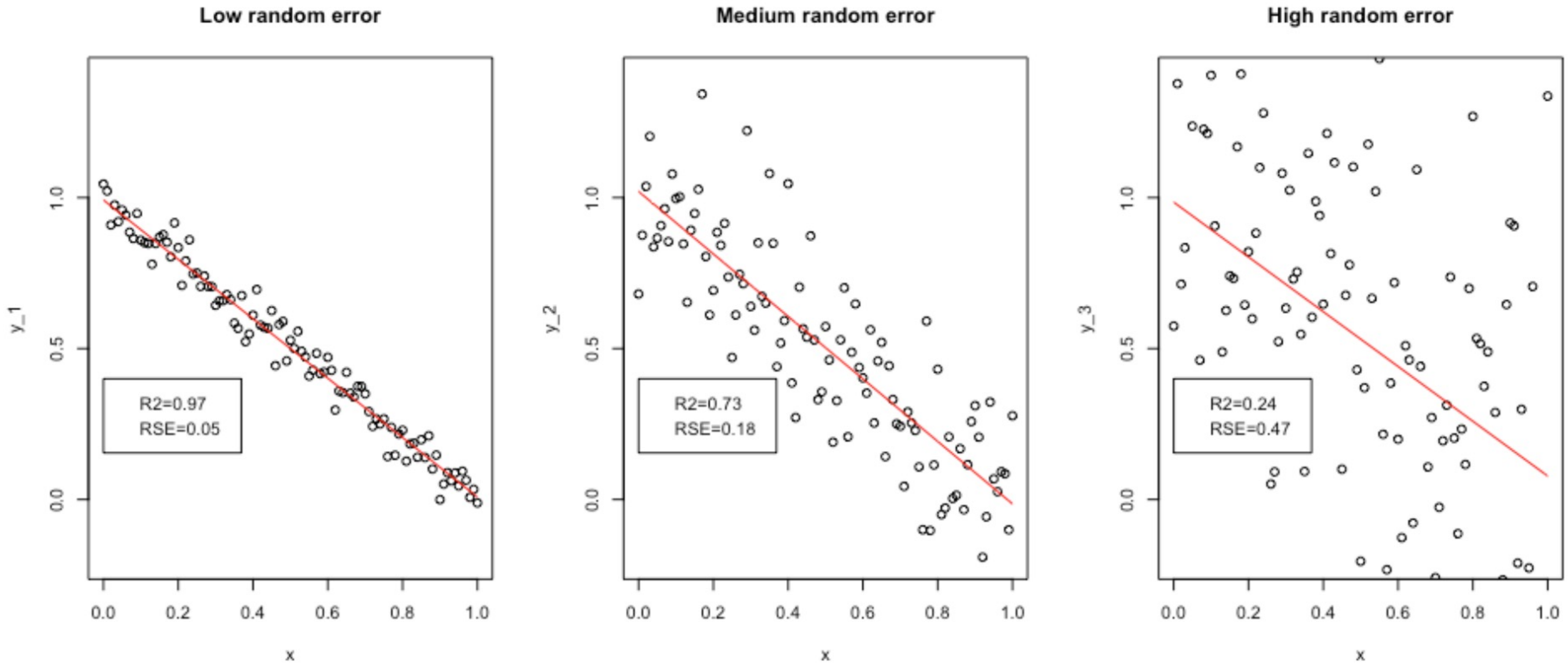
$$\mathrm{RSE} = \sqrt{\frac{1}{N-(p+1)}\sum_{i=1}^{N}(\varepsilon^{i})^2} = \sqrt{\frac{1}{N-p-1}RSS}$$

- It measures the 'corrected' standard deviation of the residuals by the degrees of freedom (DoF):
  - number of samples – number of parameters $\beta_0, \beta_1, \dots, \beta_p$

- It is in the units of the target variable: hard to assess what number is a good fit.

How to evaluate a linear regression model?

- R^2 (coefficient of determination) is a unitless metric (typically between 0 and 1) different than RSE.



How to evaluate a linear regression model?

- R^2 measures the proportion of the variation in $\mathbf{Y}$ which can be explained using $\mathbf{X}$.

- How better is our model when compared to a simple averaging model (baseline) that always outputs $\bar{y}$, independently of x?

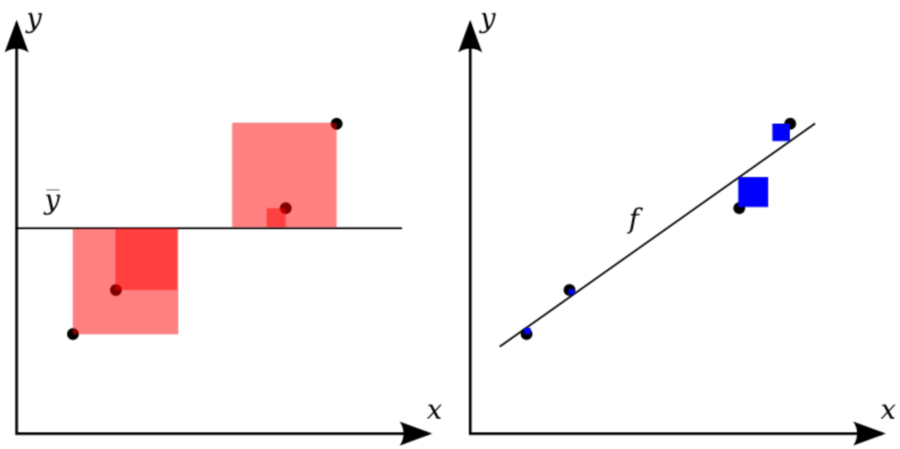
$$R^2 = 1 - \frac{\text{RSS}}{\text{TSS}}$$

- Residual sum of squares (RSS):

$$\text{RSS} = \sum_{i=1}^{N} \left(y^i - \hat{y}^i\right)^2$$

- Total sum of squares (TSS):

$$\text{TSS} = \sum_{i=1}^{N} \left(y^i - \bar{y}\right)^2$$



How to evaluate a linear regression model?

- How close are the estimated parameters $\widehat{\boldsymbol{\beta}}$ (on a dataset) to the true $\boldsymbol{\beta}$?
  - Standard deviation of each parameter → standard error (SE) for each parameter

$$\mathrm{SE}(\hat{\beta}_0) = \sigma \sqrt{\frac{\frac{1}{N}\sum_{i=1}^{N}(x^i)^2}{\sum_{i=1}^{N}(x^i - \bar{x})^2}} \qquad \mathrm{SE}(\hat{\beta}_1) = \sigma \sqrt{\frac{1}{\sum_{i=1}^{N}(x^i - \bar{x})^2}}$$

- Recap: $\sigma = \mathrm{std}(\varepsilon) = \mathrm{RSE}$

(residual standard error)

How to evaluate a linear regression model?

- How significant is the relationship with respect to dataset size?
  - Is an estimated relation a random artifact of the data?
  - Is there a significant relationship between x and y?

- For a model ($y = \beta_0 + \beta_1 x + \varepsilon$) we can test two hypotheses regarding the slope β_1

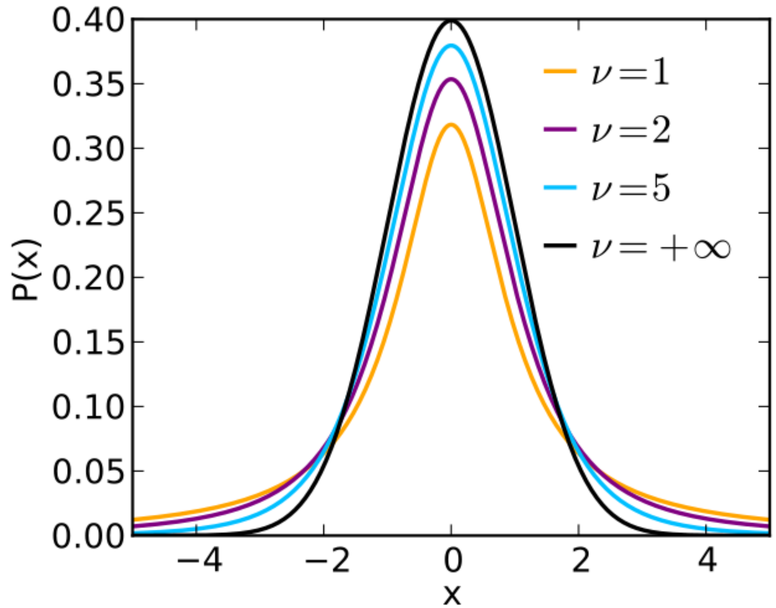
$$H_0: \beta_1 = 0$$
$$H_1: \beta_1 \neq 0$$

every x value will give the same y value and the model would be useless

- With the t-test statistic $t = \frac{\widehat{\beta}_1}{\mathrm{SE}(\widehat{\beta}_1)}$

- Reject H_0 if $t < -t_{\alpha,N-2}$ or $t > t_{\alpha,N-2}$

Student's T distribution



$P(-t_{\alpha,v} < T < t_{\alpha,v}) = 1 - 2\alpha$
v denotes DoF ($N - p$)

How to evaluate a linear regression model?

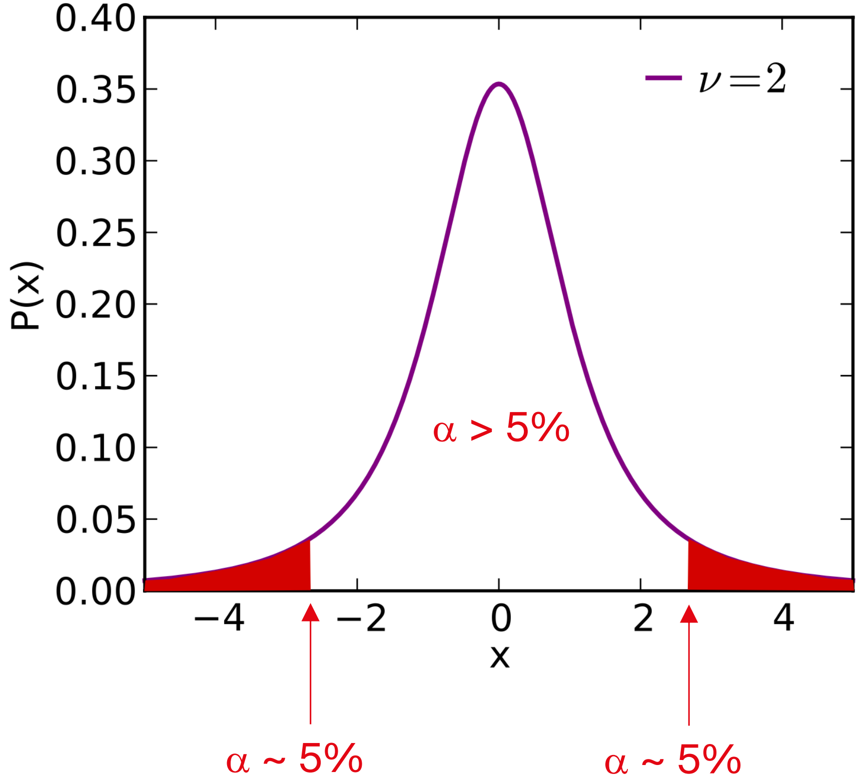
- Reject H_0 if $t < -t_{\alpha,N-2}$ or $t > t_{\alpha,N-2}$

  = being in the red area of the graph

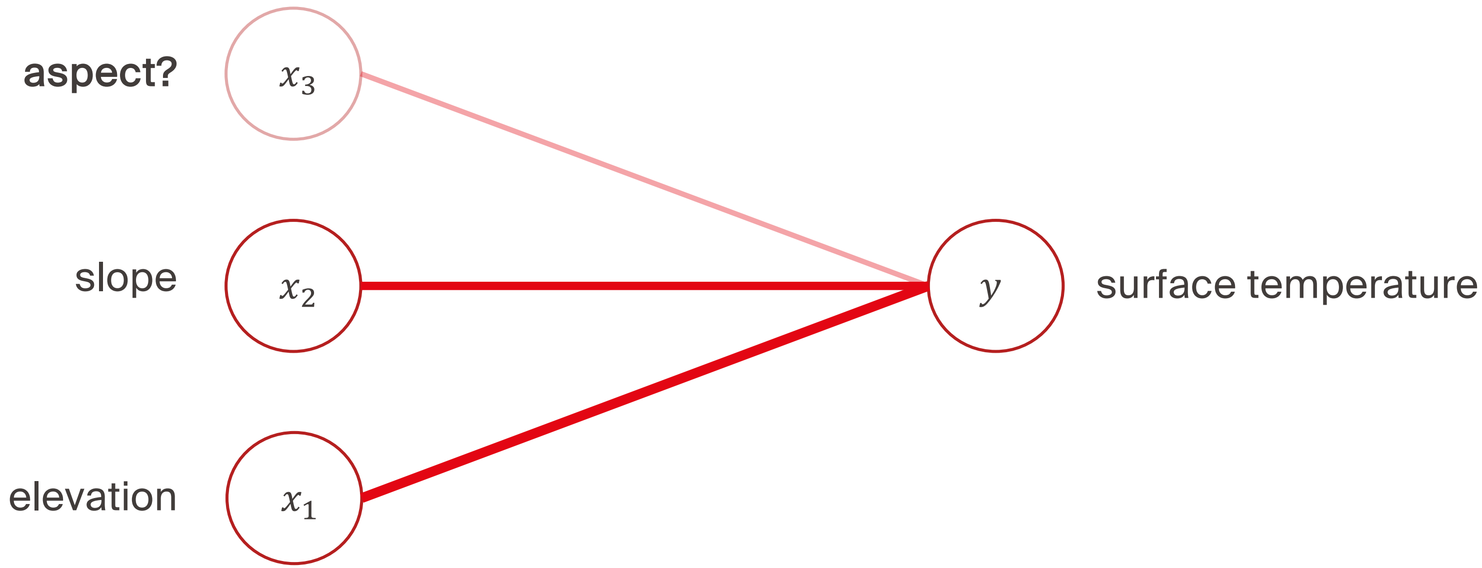
- p-value: $P(t_{\alpha,N-2} \leq t) = \alpha$

  = probability of falling in the white area of the graph, given a t score

- A high p-value (usually > 5%)
  - No significant relationship
  - Not enough data



How to select features for linear regression?



- some variables
  - can be time-consuming and costly to gather
  - can be redundant (highly correlated)

- How to automatically select a minimal subset of relevant features?
  - Based on their significance

How to select features for linear regression?

- How to automatically select a minimal subset of relevant features?
  - Sequential feature selection algorithms: remove or add relevant features based on their significance!

Criterion: smallest p-value, highest increase in R2, highest drop in model RSS compared to other predictors under consideration.

Stopping rule: number of desired features or criterion threshold

Forward Selection Algorithm

```
null model without features
while stopping criterion is not met do
    for each candidate feature do
        add feature
        fit model
        score model
    add the best feature to the model
```

Backward Selection Algorithm

```
full model with all features
while stopping criterion is not met do
    for each feature in the model do
        remove feature
        fit model
        score model
    remove the worst feature
```

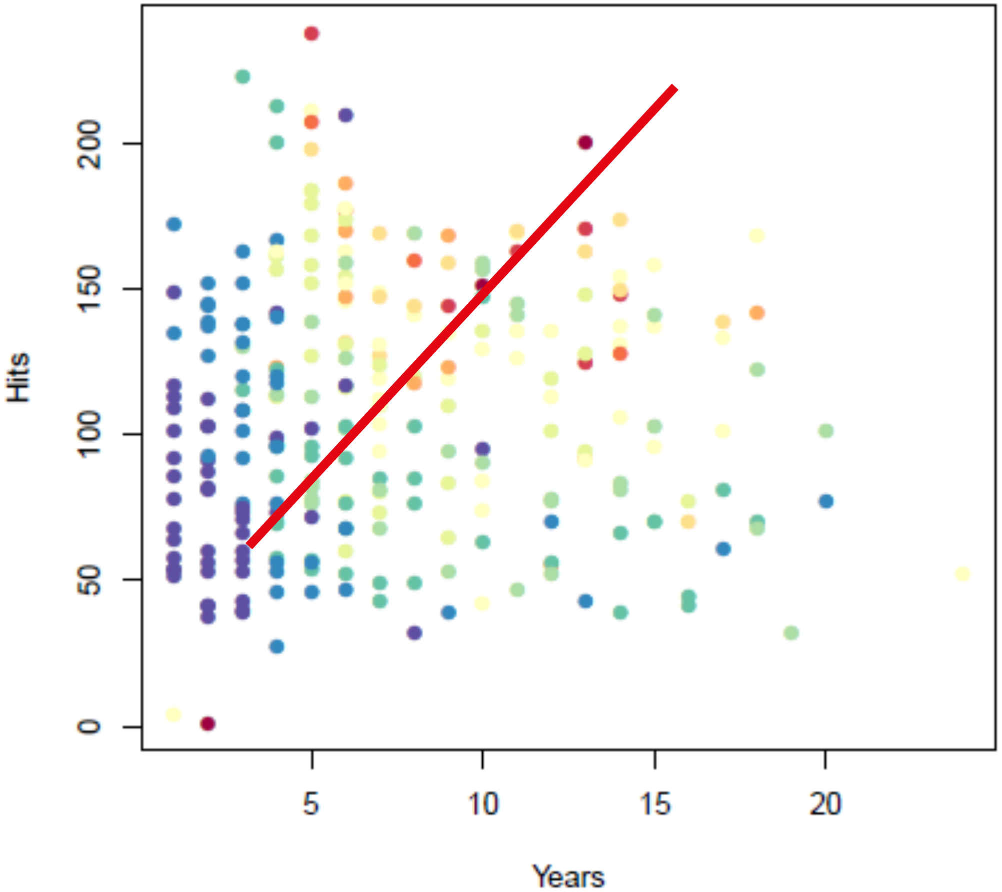
EPFL

Decision Trees for Regression

Tuia / Sumbul / Dalsasso

EPFL

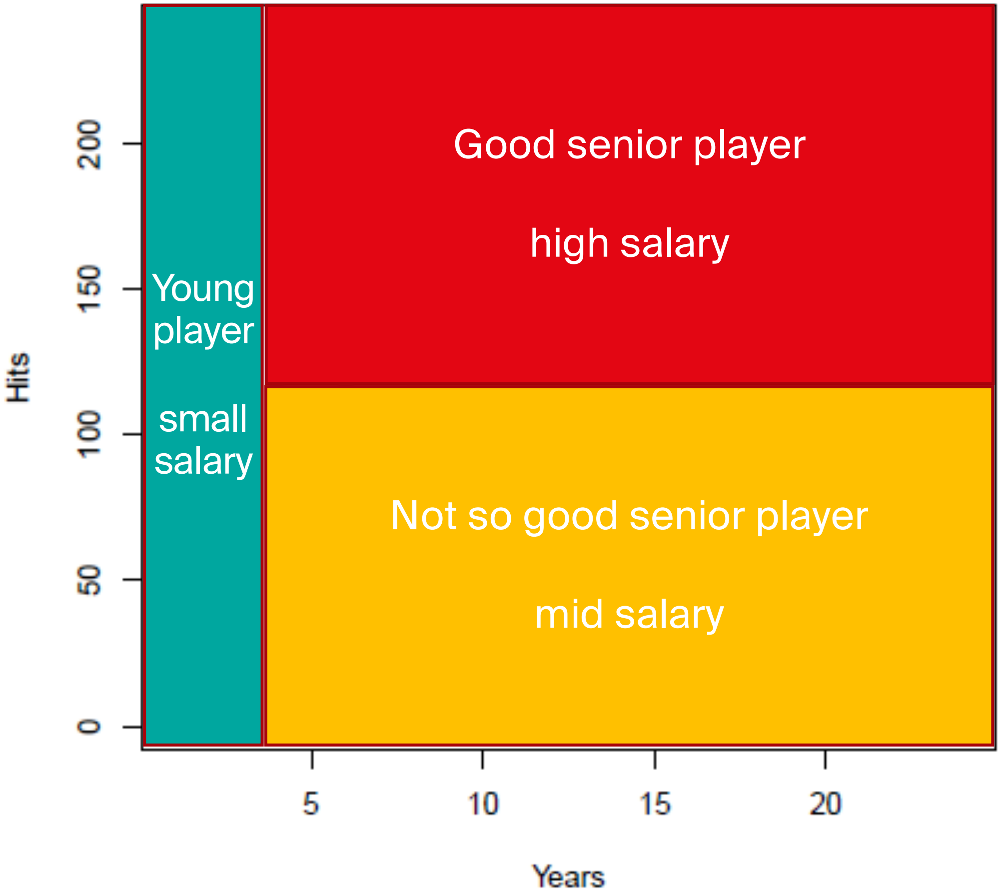
How would you predict salary from years/hits?



Salary is color-coded from low (blue, green) to high (yellow,red)

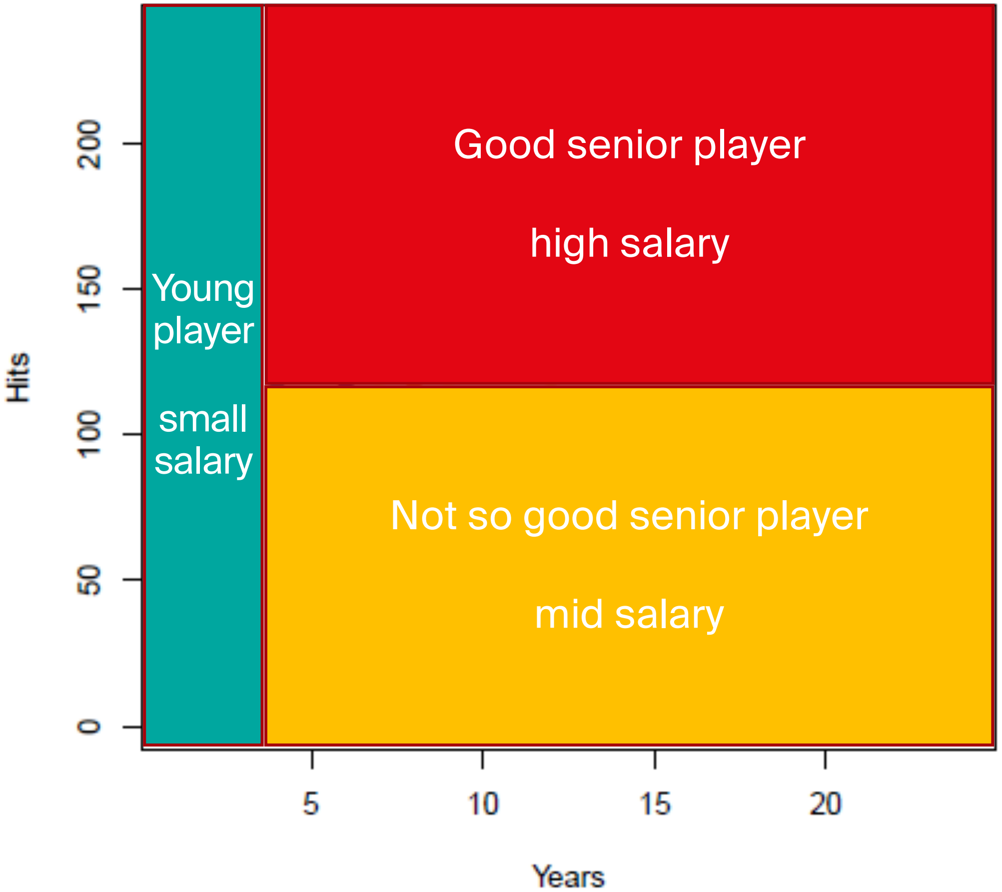
There is no linear relationship between years/hits and salary!

Segmenting the space in coherent partitions?

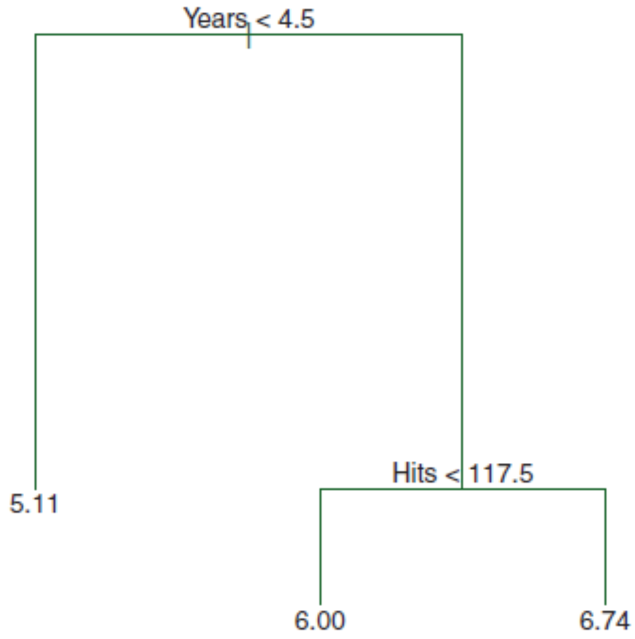


Salary is color-coded from low (blue, green) to high (yellow,red)

Segmenting the space in coherent partitions?



It becomes a decision tree!

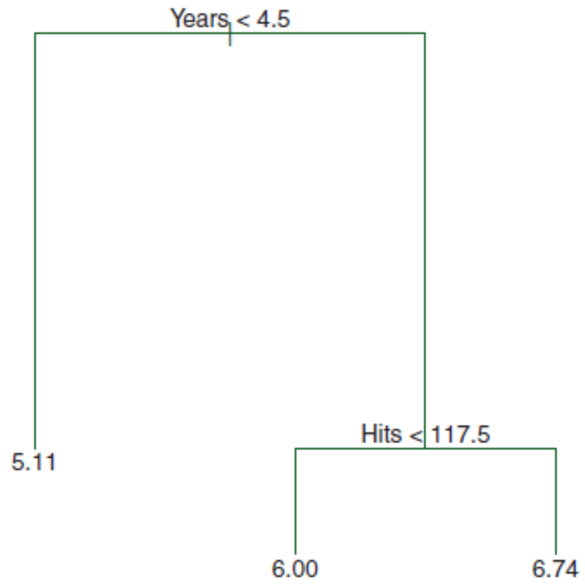


Salary is color-coded from low (blue, green) to high (yellow,red)

Stratifying (segmenting) predictor space

- A decision tree is an interpretable model in which the final output is based on a series of comparisons of the values of predictors against threshold values.
  - Nonlinear by design!
  - Hierarchical
  - Non-parametric

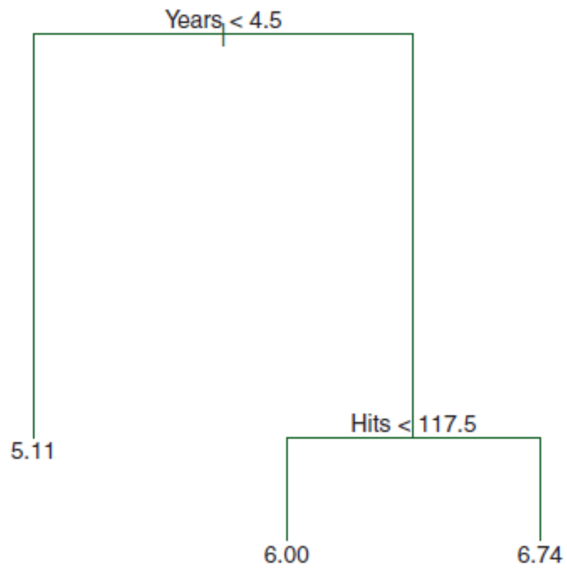
- It basically segments the input space by using a supervised rule:
  - "if I divide there, would the two resulting segments be clearer about the quantity being predicted"?

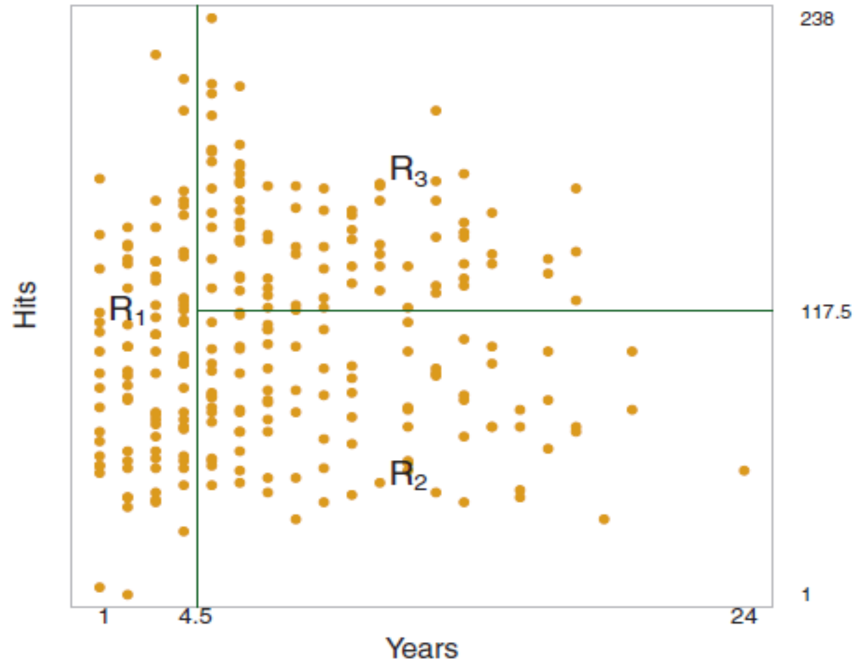


Stratifying (segmenting) predictor space

- Graphically, decision trees can be represented by a flow chart.

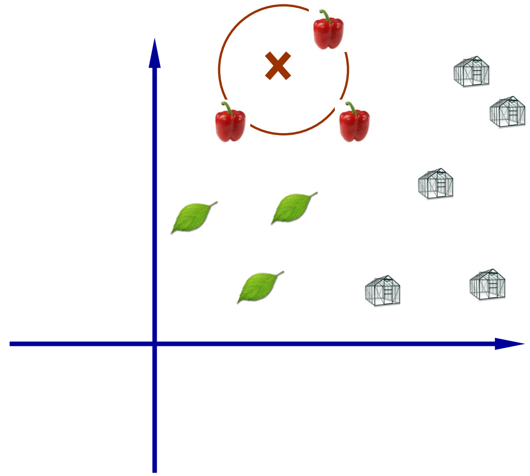
- Geometrically, the model partitions the feature space, where each region is assigned a response value based on the samples of the region.



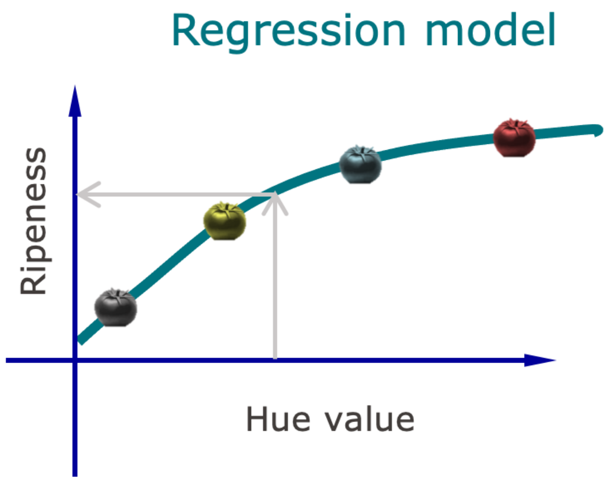


Stratifying (segmenting) predictor space

- Tree-based methods are usually used for classification

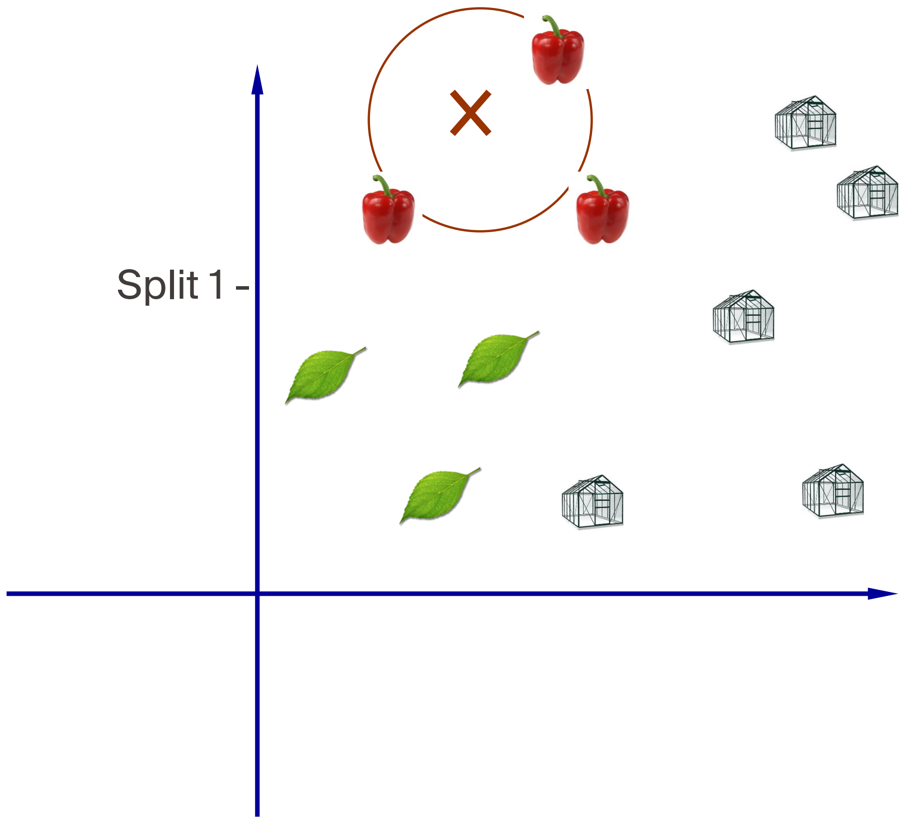


- Their concept translates well to regression problems too



In the classification case

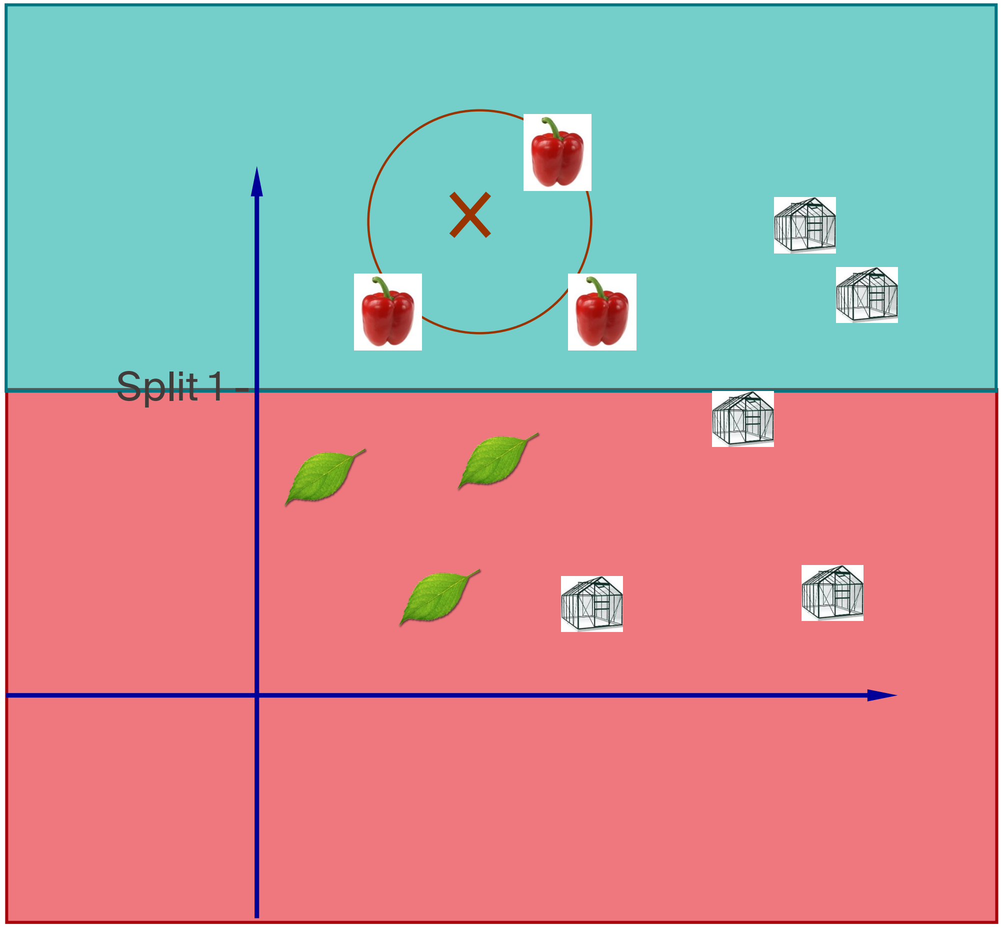
- You can find the nonlinear solution in 3 splits.



Feature Space

In the classification case

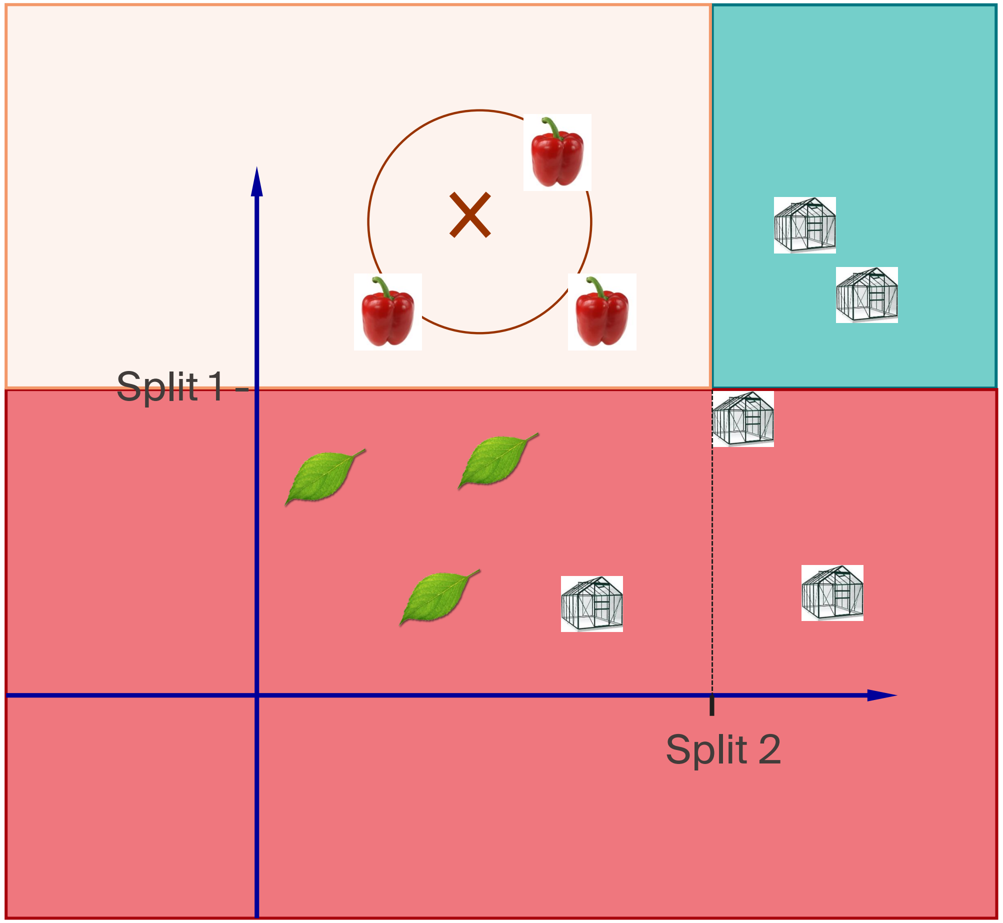
- You can find the nonlinear solution in 3 splits.



Feature Space

In the classification case

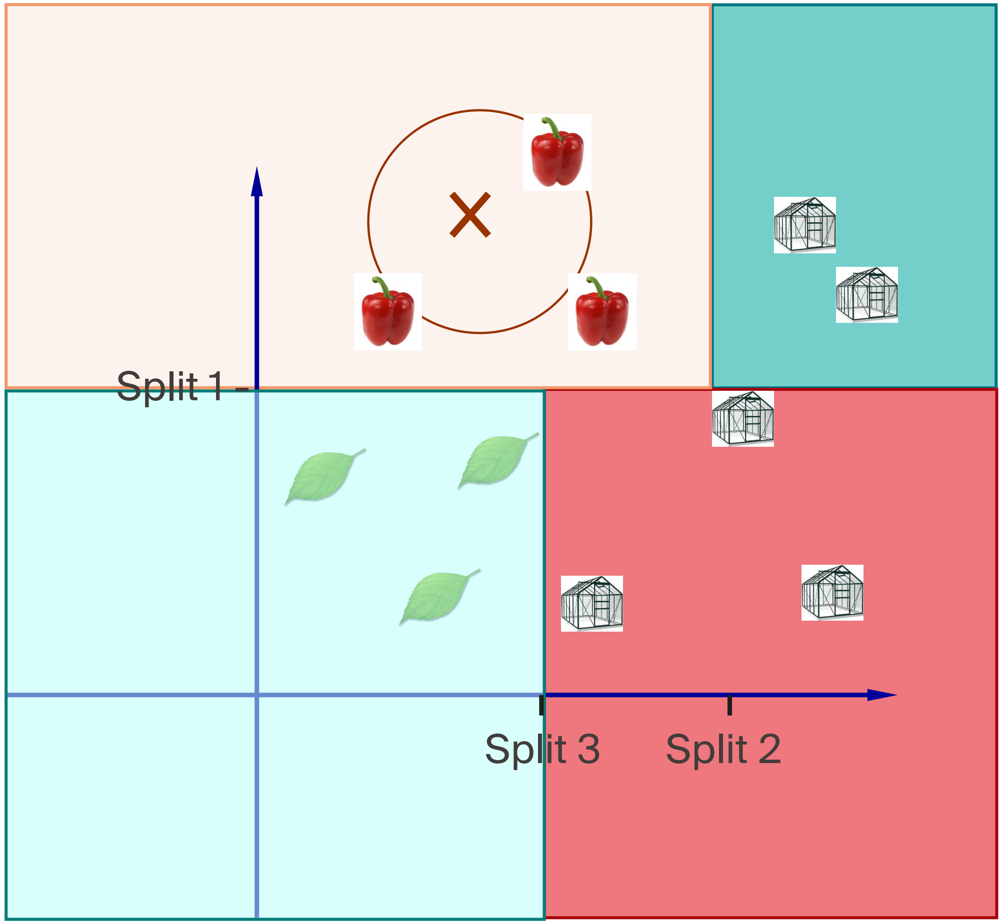
- You can find the nonlinear solution in 3 splits.
- (with 2 you would get the bell pepper right)



Feature Space

In the classification case

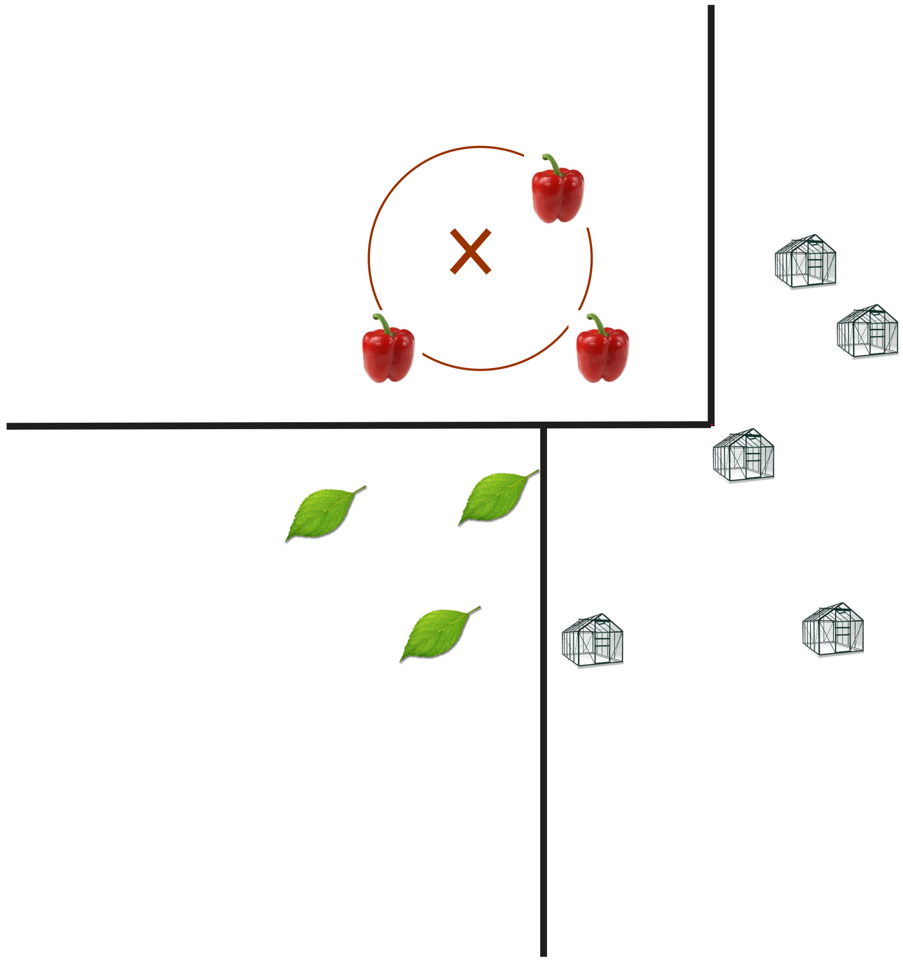
- You can find the nonlinear solution in 3 splits.



Feature Space

In the classification case

- You can find the nonlinear solution in 3 splits.

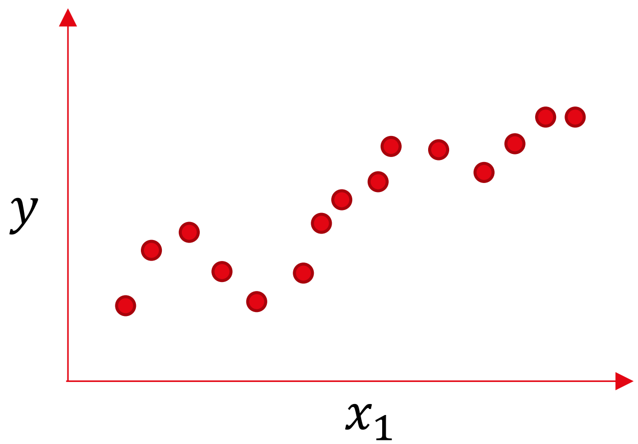


Feature Space

How to build a tree?

- By constructing regions of the feature space

- For regression
  - Use mean of observations in each region



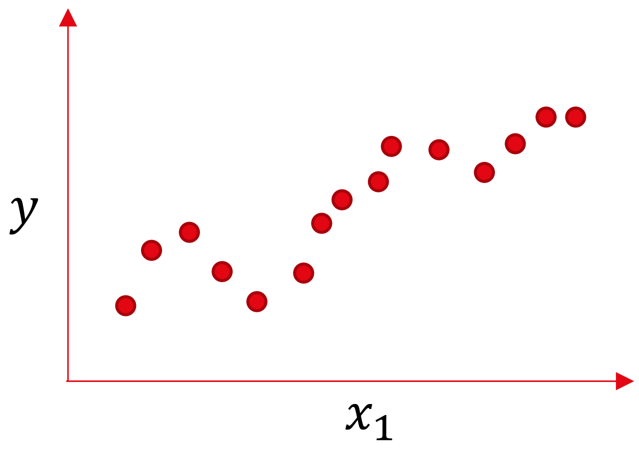
- For classification
  - Majority voting in each region

How to build a tree?

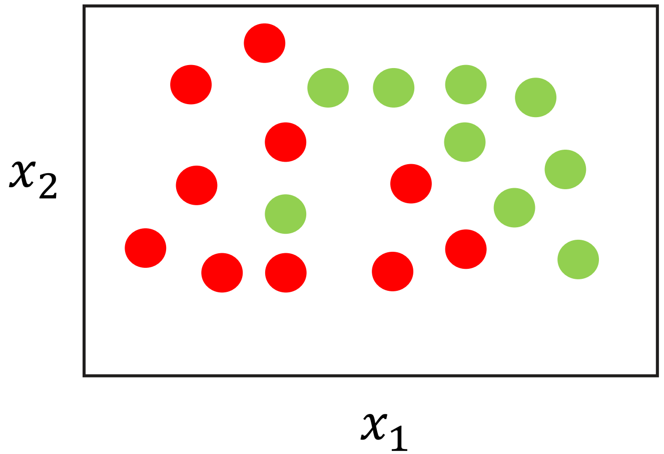
*careful! This regression example is in 1D in contrast to the baseball example before or the classification one on the right, which are both 2D

- By constructing regions of the feature space

- For regression*
  - Use mean of observations in each region



- For classification
  - Majority voting in each region

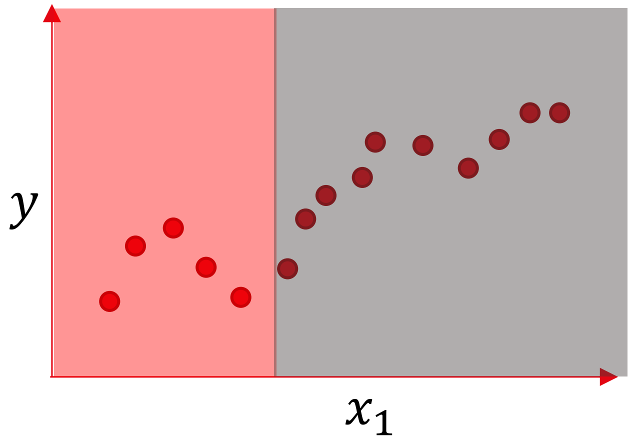


Tuia / Sumbul / Dalsasso

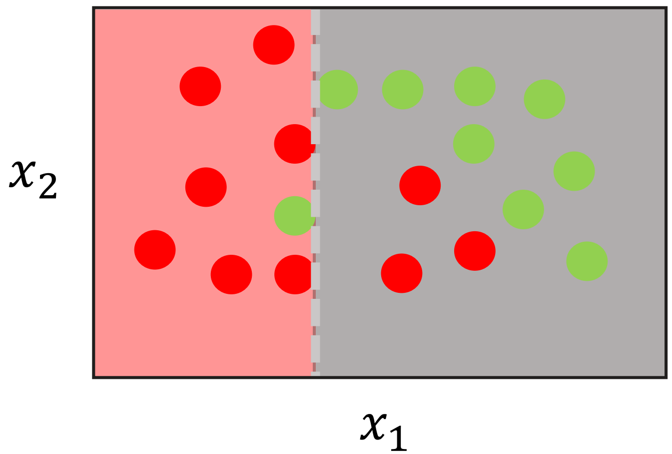
How to build a tree?

- By constructing regions of the feature space

- For regression
  - Use mean of observations in each region



- For classification
  - Majority voting in each region

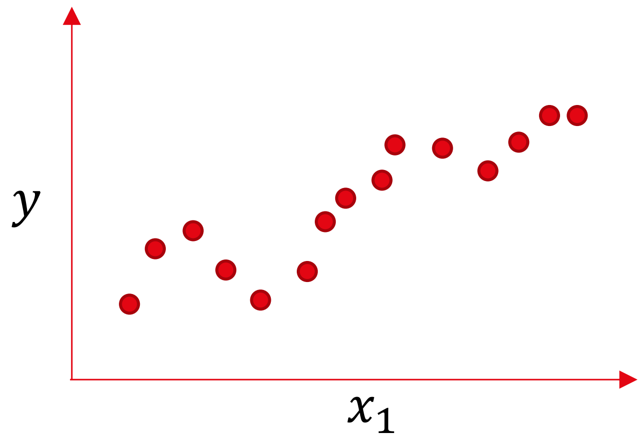


Constructing regions for a regression tree

Find the regions $R_1, R_2, \ldots, R_j$ minimizing residual sum of squares (RSS):

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

With $\bar{y}_{R_j}$ being mean response for training samples in region R_j

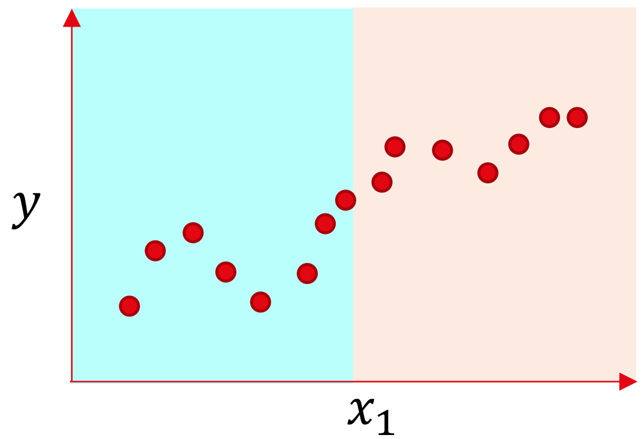


Constructing regions for a regression tree

Find the regions $R_1, R_2, \ldots, R_j$ minimizing residual sum of squares (RSS):

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

With $\bar{y}_{R_j}$ being mean response for training samples in region R_j

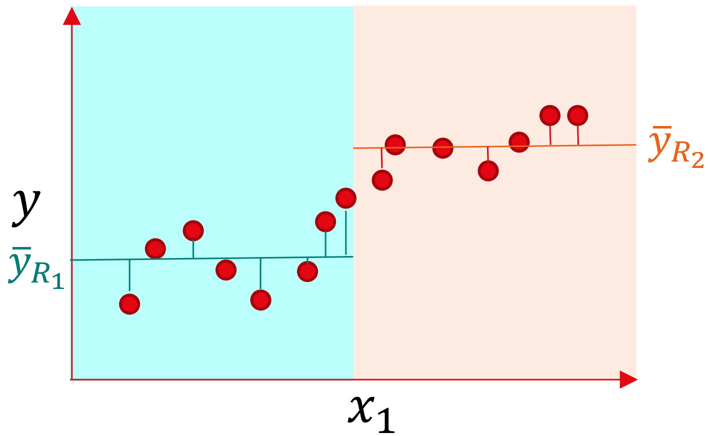


Constructing regions for a regression tree

Find the regions $R_1, R_2, \dots, R_j$ minimizing residual sum of squares (RSS):

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

With $\bar{y}_{R_j}$ being mean response for training samples in region R_j



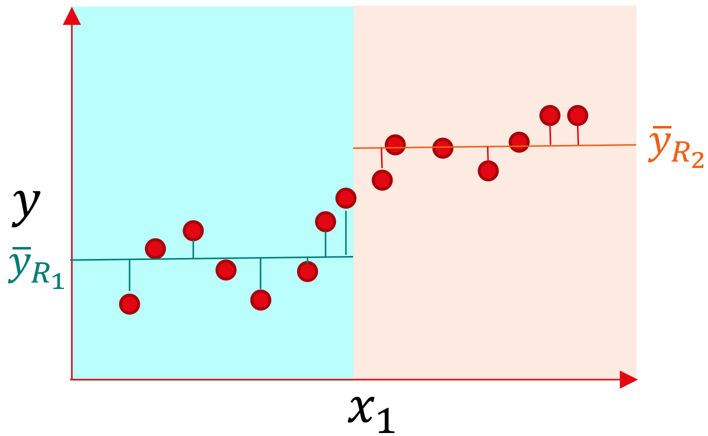
Constructing regions for a regression tree

Find the regions $R_1, R_2, \ldots, R_j$ minimizing residual sum of squares (RSS):

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

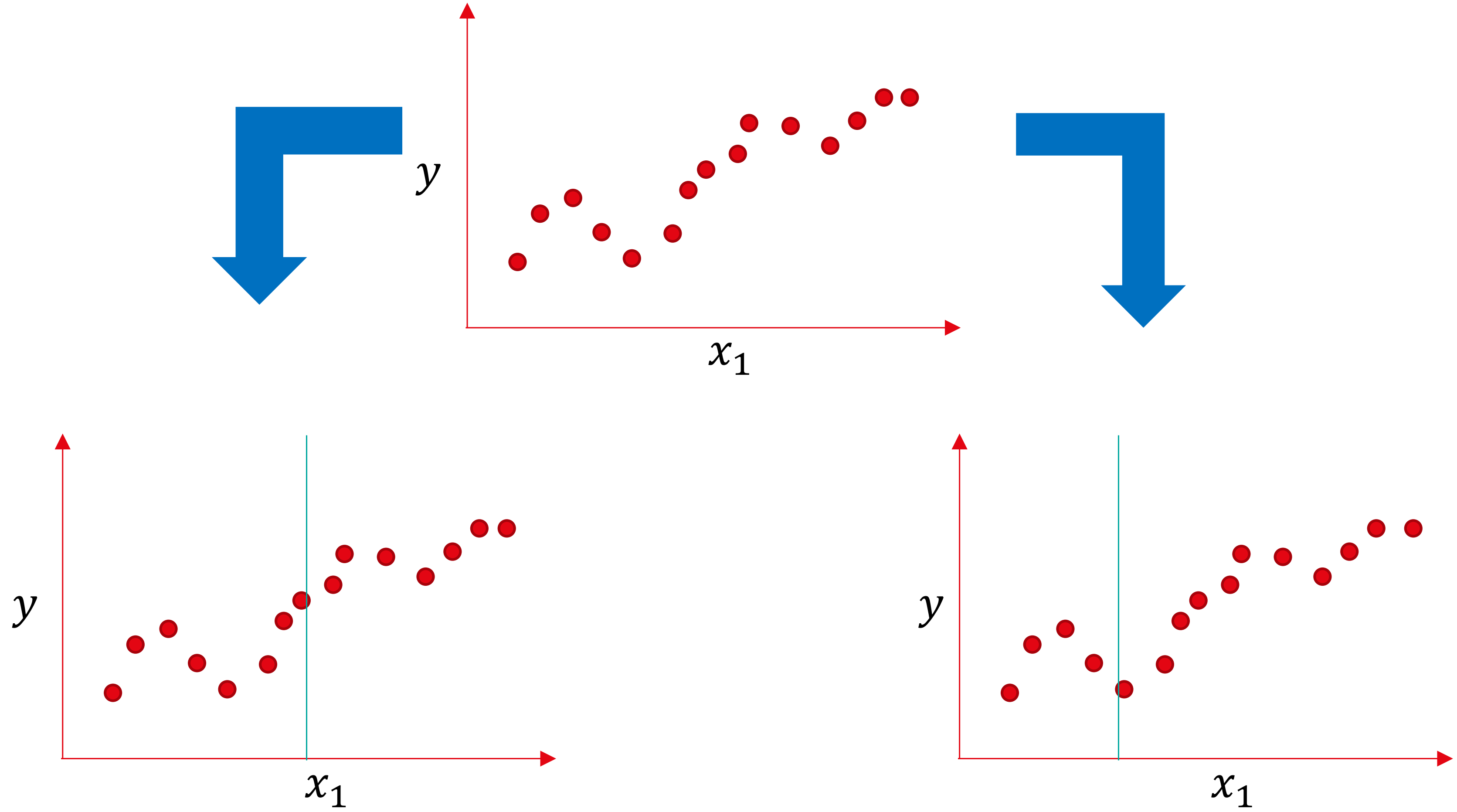
With $\bar{y}_{R_j}$ being mean response for training samples in region R_j

All samples in the green part will be Predicted as $\bar{y}_{R_1}$

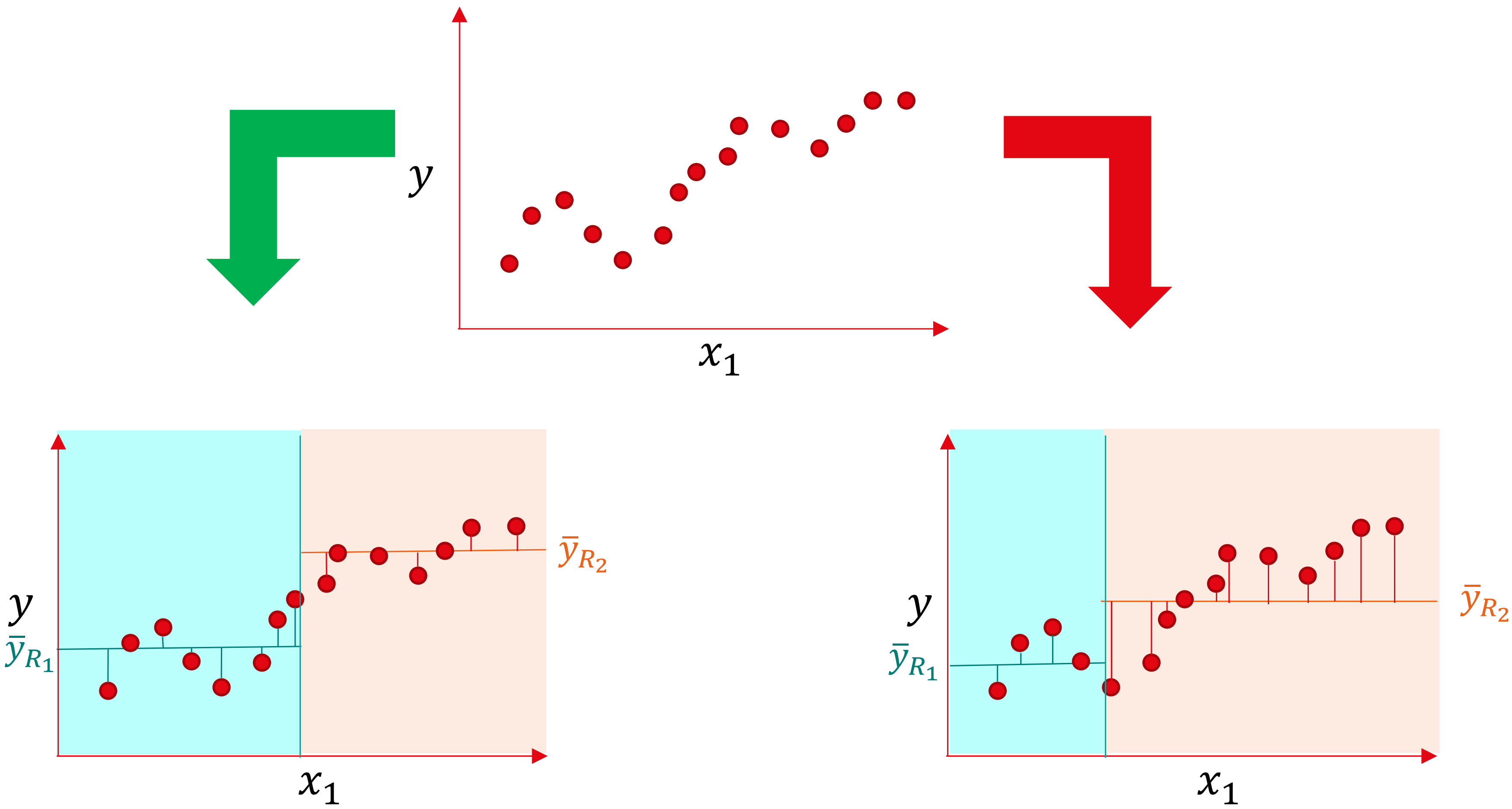


All samples in the orange part will be Predicted as $\bar{y}_{R_2}$

Example of 2 different splits



Example of 2 different splits



How to construct regions?

- Recursive binary splitting: Top-down, greedy approach
  - Start at top of tree, successively split predictor space
  - Best split is made at the current step
  - Not looking forward, to find a split that at future step might give a better result
  - Finish when reaching a stopping condition (e.g., each leaf has fewer than some fixed number of instances)

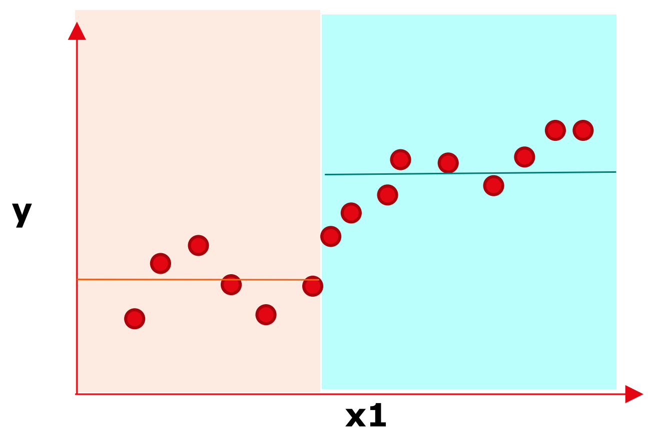
- Why not to consider every possible partition of the feature space?
  - Computationally infeasible (NP-hard)!

Constructing regions for a regression tree

Find the regions $R_1, R_2, \dots, R_j$ minimizing residual sum of squares (RSS):

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

With $\bar{y}_{R_j}$ being mean response for training samples in region R_j

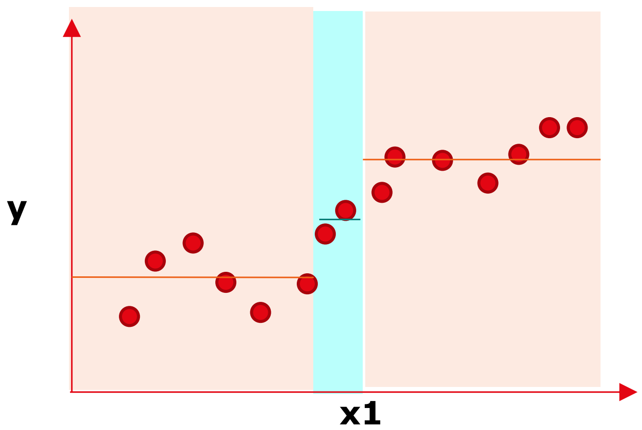


Constructing regions for a regression tree

Find the regions $R_1, R_2, \dots, R_j$ minimizing residual sum of squares (RSS):

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

With $\bar{y}_{R_j}$ being mean response for training samples in region R_j

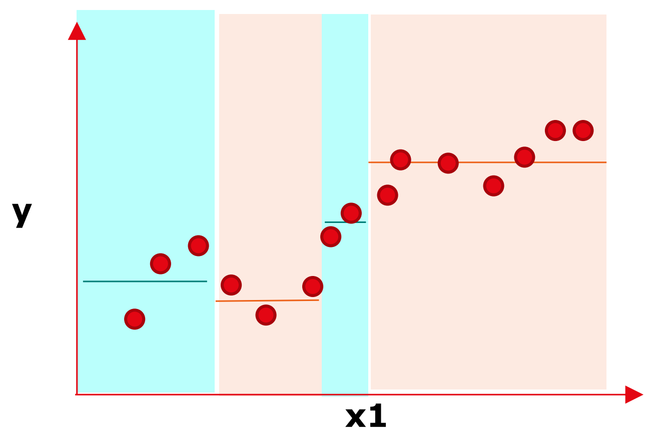


Constructing regions for a regression tree

Find the regions $R_1, R_2, \dots, R_j$ minimizing residual sum of squares (RSS):

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

With $\bar{y}_{R_j}$ being mean response for training samples in region R_j

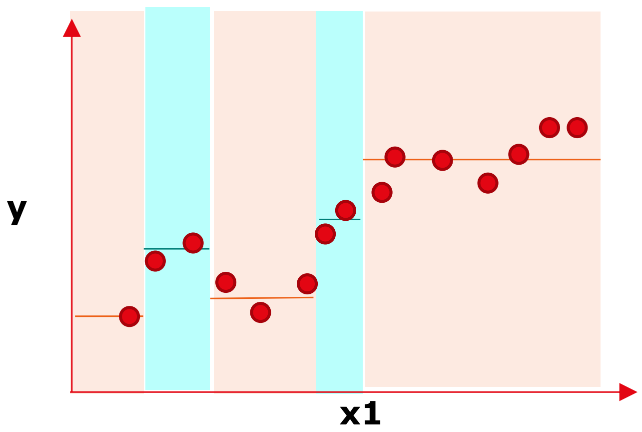


Constructing regions for a regression tree

Find the regions $R_1, R_2, \ldots, R_j$ minimizing residual sum of squares (RSS):

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

With $\bar{y}_{R_j}$ being mean response for training samples in region R_j

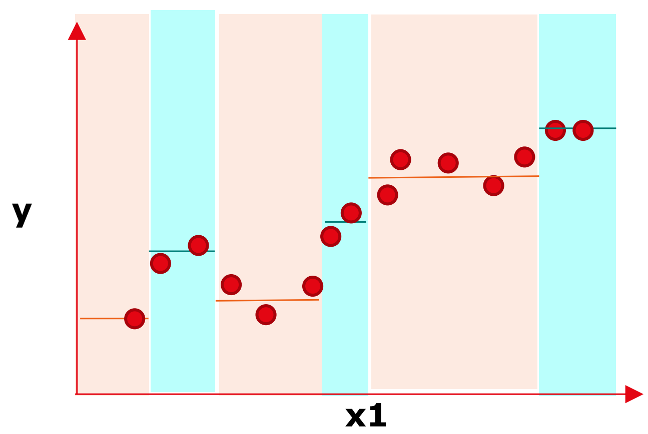


Constructing regions for a regression tree

Find the regions $R_1, R_2, \ldots, R_j$ minimizing residual sum of squares (RSS):

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

With $\bar{y}_{R_j}$ being mean response for training samples in region R_j

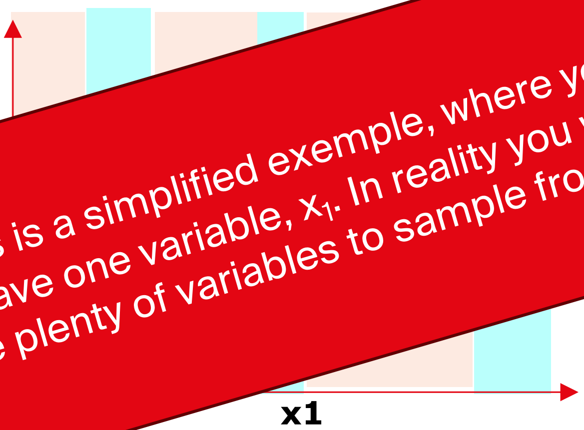


Constructing regions for a regression tree

Find the regions $R_1, R_2, \ldots, R_j$ minimizing residual sum of squares (RSS):

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

With $\bar{y}_{R_j}$ being mean response for training samples in region R_j



Ok, this is a simplified exemple, where you only have one variable, x_1. In reality you will have plenty of variables to sample from!

Why will this procedure (may) lead to overfitting?

- (Too) complex tree will be preferred
- Solution 1: early stopping
- Solution 2: pruning

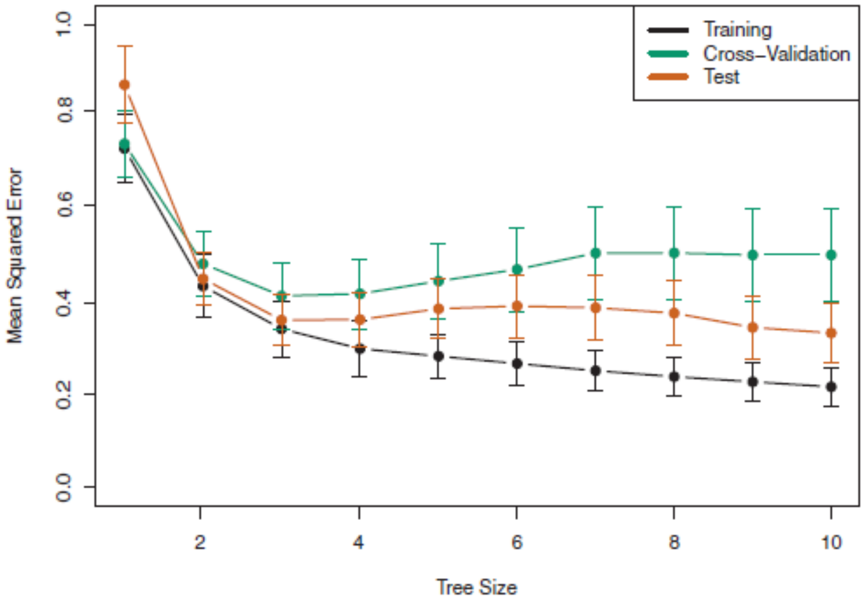
Pruning

- First, we grow a very large tree T_0, and then prune it back to obtain a subtree.

$$\text{for } T \subset T_0 \text{ minimize } \sum_{m=1}^{|T|} \sum_{i \in R_m} (y^i - \bar{y}_{R_m})^2 + \alpha |T|$$

predictor space in the mth leaf

Number of leaves (terminal nodes)



Bagging for Regression

Tuia / Sumbul / Dalsasso

From tree to forest

- A single decision tree can overfit
  - low bias (less assumptions, high flexibility)
  - high variance (data sensitivity)

- A single decision tree can suffer from high variance (data sensitive)
  - for different training sets, decisions can be quite different

- The concept of bagging is meant to reduce such variance by building a committee of models.

- Random forests (RF) use it.

Bagging

- Let's say a single decision tree has an output $\widehat{\mathbf{Y}}$ with variance σ^2

- If we repeat the modeling with n independent trials, we get n models $\widehat{\mathbf{Y}}^1, \widehat{\mathbf{Y}}^2, \widehat{\mathbf{Y}}^3, \ldots, \widehat{\mathbf{Y}}^n$ each with variance σ^2.

- According to the central limit theorem, the variance of their average has variance σ^2/n.

- Averaging independent models reduces variance!

Bagging: bootstrap aggregation

- In practice, we train B different methods with subsets of the data, and then average them out:

$$\hat{f}_{\text{bag}}(x) = \frac{1}{B}\sum_{b=1}^{B} \hat{f}^{*b}(x)$$

- Even more in practice, we can't have truly independent subsets

- We resample parts of the data and use them in each model training.
  - random sampling with replacement

Random Forests for Regression

Tuia / Sumbul / Dalsasso

Random Forests

- Bagging trees with more randomization

- For each split in a tree, we also consider a random subset of features (typically $\sqrt{p}$ if you have p features to start with)

- As before, we average the predictions of the B trees

$$\hat{f}_{\mathrm{RF}}(x) = \frac{1}{B}\sum_{b=1}^{B} \hat{f}^{*b}(x)$$

Random Forests

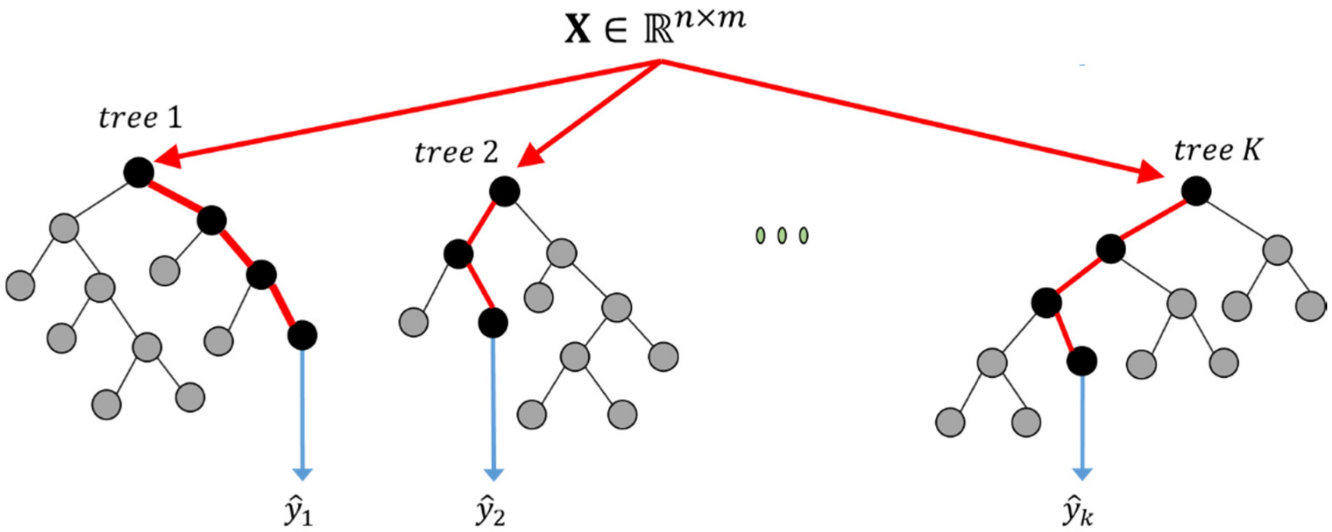
- For each tree:
  - Select a subset of the data

  - For each node:
    - Select some of the variables
    - Calculate some split values for those variables
    - Select the best partition
    - Split the data points into two groups, which become new nodes

  - Predict the response for this tree

- Take the average prediction across all trees

Random Forests



$$\hat{y} = \frac{1}{K}\sum_{k=1}^{K} \hat{y}_k$$

In summary

- Today we saw two approaches to (non)linear regression

  - Linear regression, uni-and multivariate. The most popular,
    - but “just” linear, you need to design good nonlinear features

  - Decision trees-based regression. Nonlinear by design,
    - By partitioning the predictors space into good average approximations
    - Repeating over and over

- Remember that all these ML approaches need data to train, the more the better