



LECTURE 3

MACHINE LEARNING ON DATA REPRESENTATIONS

Learning on fixed representations

PRESS **→** OR SPACE. 

Learning Outcome Roadmap

Learning outcome	Lecture	Block
☐ Explain how heterogeneous datasets align to a common coordinate support and form a geospatial data cube.	Lecture 3	Geospatial Data
☐ Represent locations as fixed feature vectors extracted from a geospatial data cube.	Lecture 3	Geospatial Data
☐ Explain how linear regression maps a fixed representation to a continuous target.	Lecture 3	Linear Models
☐ Explain how logistic regression maps a fixed representation to categories with a linear decision boundary.	Lecture 3	Linear Models
☐ Explain how decision trees partition a fixed representation space into nonlinear decision regions.	Lecture 3	Decision Trees
☐ Explain how random forests combine trees for robust predictions from fixed representations.	Lecture 3	Decision Trees

Geospatial Data Cubes

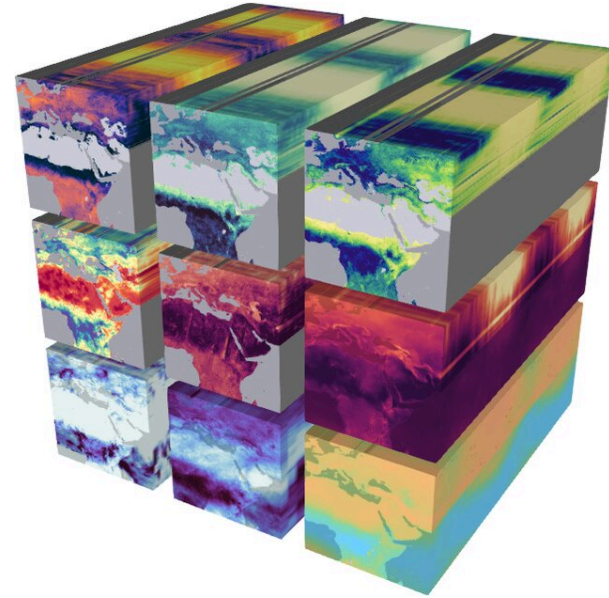


Image by [Miguel D. Mahecha](#)

Recap: Geospatial Data

- satellite imagery
- thematic rasters such as land cover
- vector geometries and attributes
- model and reanalysis fields
- observations collected at different times and scales

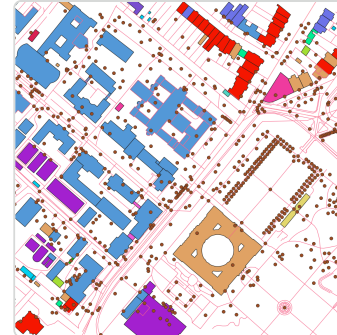
Each source describes locations on its own spatial and temporal support.



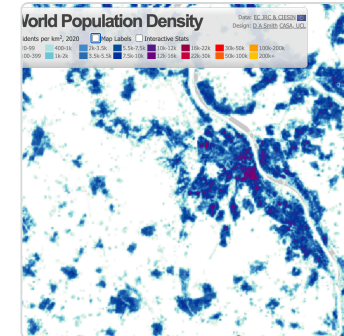
Satellite imagery (Bonn, Sentinel-2)



Thematic raster (land cover, Amsterdam)

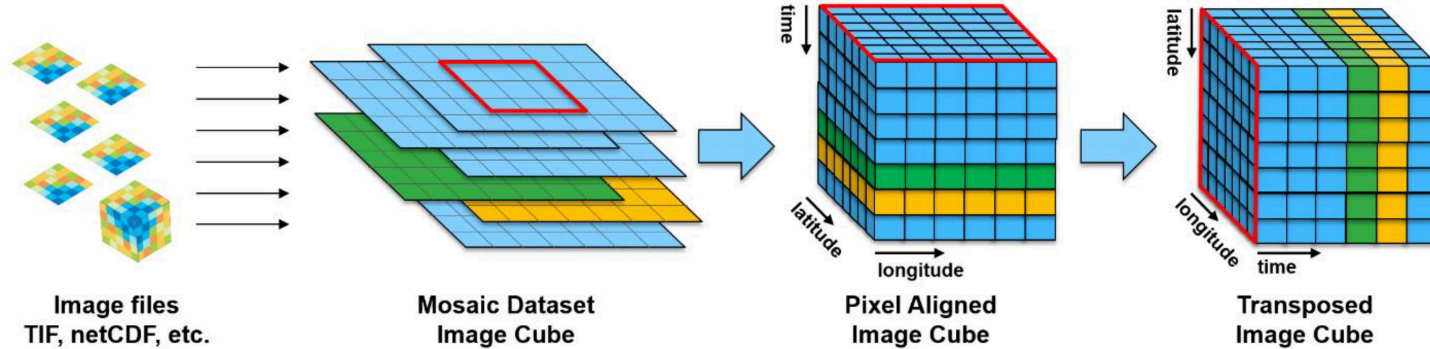


Vector geometries (OpenStreetMap)



Model / reanalysis field (population density)

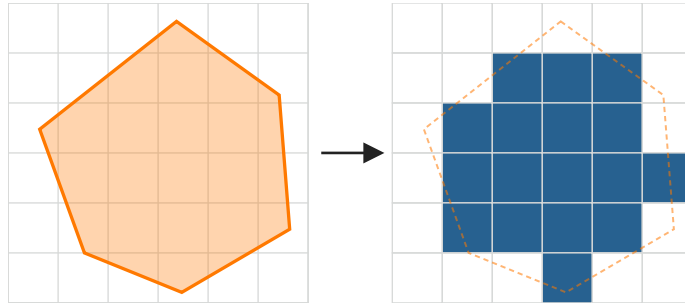
Datacubes align Data Spatially and Temporally



Source: Kopp et al. (2019), [Achieving the Full Vision of Earth Observation Data Cubes](#), Data 4(3), 94.

Rasterization and Interpolation

Rasterization

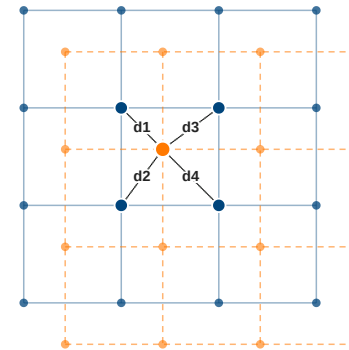


Open-source tools:

`gdal_rasterize` (CLI)

`rasterio.features.rasterize` (Python)

Interpolation



● source grid ● target grid ● reference point

Open-source tools:

`gdalwarp` / `gdal_translate` (CLI)

`rasterio.warp.reproject` (Python)

Bilinear

2×2 neighborhood

$$\hat{z} = \frac{\sum_{i=1}^4 z_i / d_i}{\sum_{i=1}^4 1 / d_i}$$

Cubic

4×4 neighborhood

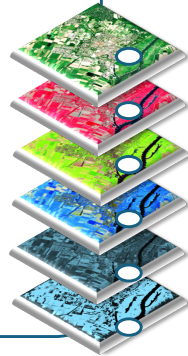
$$\hat{z} = \frac{\sum_{i=1}^{16} z_i / d_i^3}{\sum_{i=1}^{16} 1 / d_i^3}$$

From Data Cube to Feature Vector

physical data $d[\lambda, \phi]$ we
can **measure at a location**

all geodata layers

d_1 soil moisture
 d_2 reflectance
 d_3 temperature
 d_4 elevation
 d_5 population
 d_6 ecosystem
 d_7 precipitation



Sampling all p variables at location i produces a fixed feature vector:

$$\mathbf{x}_i = [x_{i1}, x_{i2}, \dots, x_{ip}]$$

For example:

$$\mathbf{x}_i = [\text{elevation, slope, spectral bands,} \\ \text{temperature, building density}]$$

Repeating the extraction at n locations creates a feature matrix:

$$\mathbf{X} \in \mathbb{R}^{n \times p}$$

features $\mathbf{x}_i$ → machine-learning model → prediction $\hat{y}_i$

Training target y_i : kept separate from the input features.

Dataset

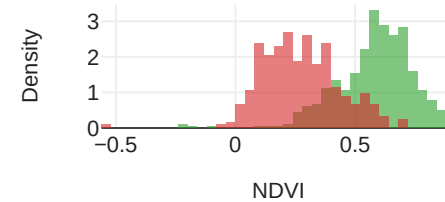
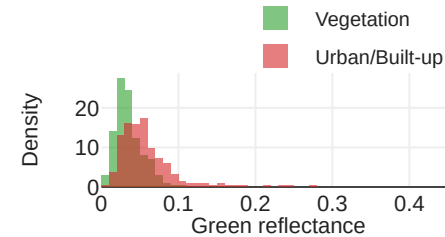
Raw bands

Pt	Red	Green	Blue	NIR	Land Cover
Pt1	0.08	0.11	0.09	0.42	Vegetation
Pt2	0.21	0.19	0.15	0.05	Built-up
⋮	⋮	⋮	⋮	⋮	⋮

Engineered feature

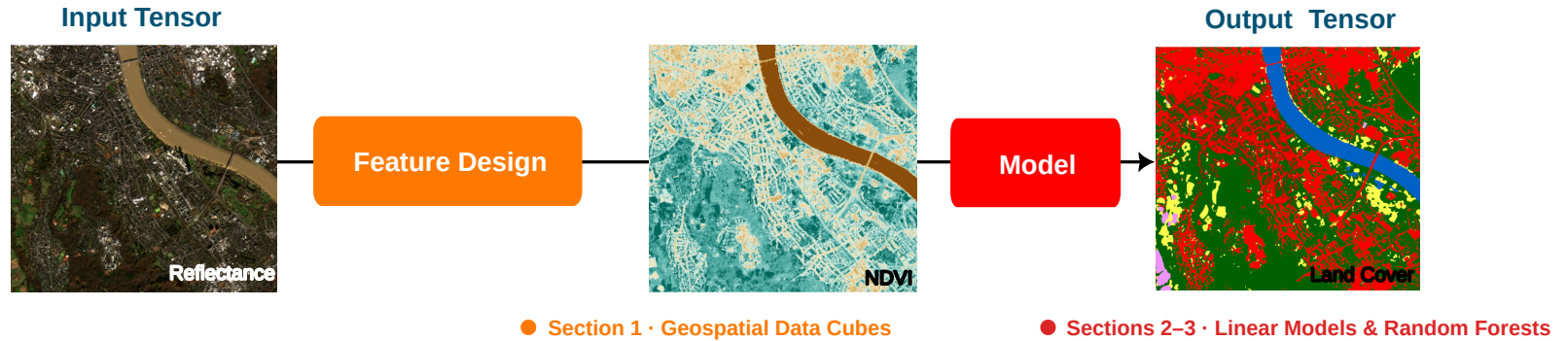
$$\text{NDVI} = \frac{\text{NIR} - \text{Red}}{\text{NIR} + \text{Red}}$$

Pt	NDVI	Land Cover
Pt1	0.68	Vegetation
Pt2	-0.62	Built-up
⋮	⋮	⋮



Classic Machine Learning Pipeline

Good, hand-selected features combined with a suitable model determine the quality of the result.



Linear Regression

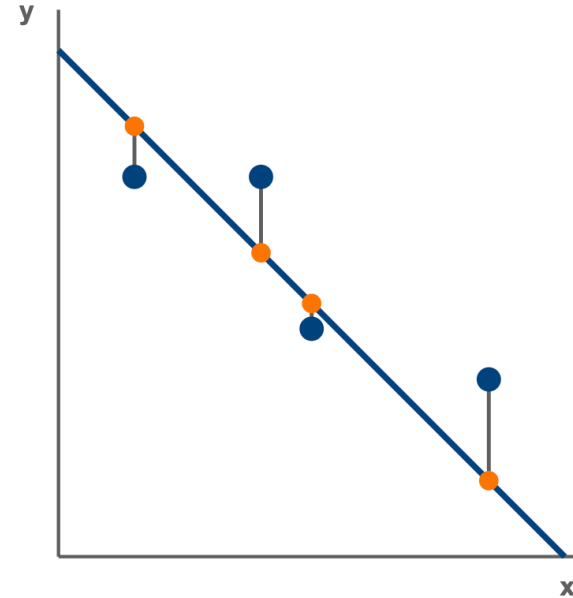


Figure: M. Rußwurm & D. Tuia, *Linear Regression*, ENV-408 course materials, EPFL, 2023.

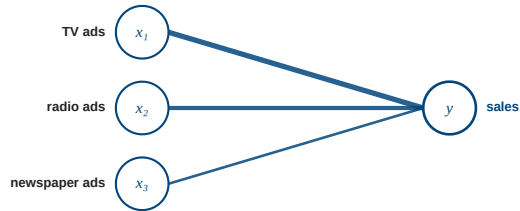
Why Linear Regression?

- a simple, classical approach to supervised learning
- predicts a **continuous** response from one or more predictors
- around for a long time, yet still one of the most used tools in science
- a strong, interpretable baseline — and a stepping stone to more complex models

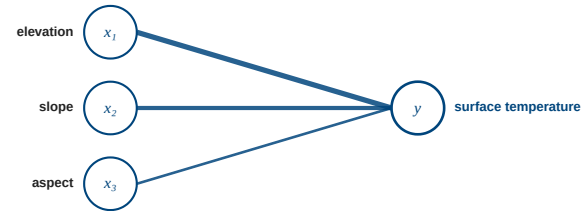
Learning linear regression helps understand more complex methods, and it is practically useful and easy to implement for many real-world problems.

Measuring Relationships Between Variables

Classic example



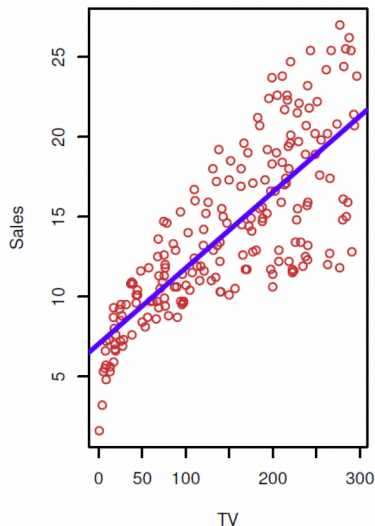
Geospatial example



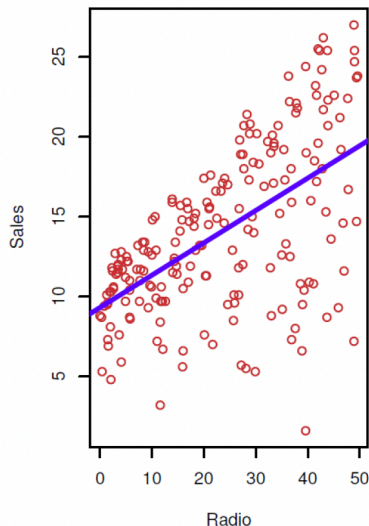
Linear regression models how one or more predictor variables (features) relate to a continuous target — exactly the fixed representations built in the previous section.

A Closer Look at the Data

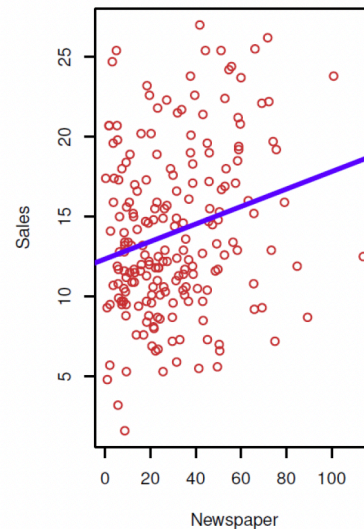
$$\mathcal{D} = \{((x_{1,i}, x_{2,i}, x_{3,i}), y_i)\}_{i=1}^N$$



$x_1 \mapsto y$ (TV $\rightarrow$ sales)



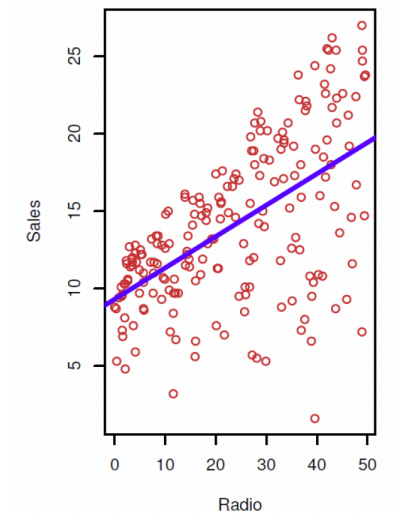
$x_2 \mapsto y$ (radio $\rightarrow$ sales)



$x_3 \mapsto y$ (newspaper $\rightarrow$ sales)

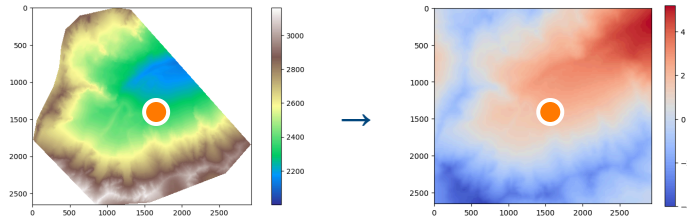
Linear Regression Answers Important Questions

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



The same questions apply to our geospatial example: is there a relationship between elevation and surface temperature — and how strong is it?

Univariate Linear Regression



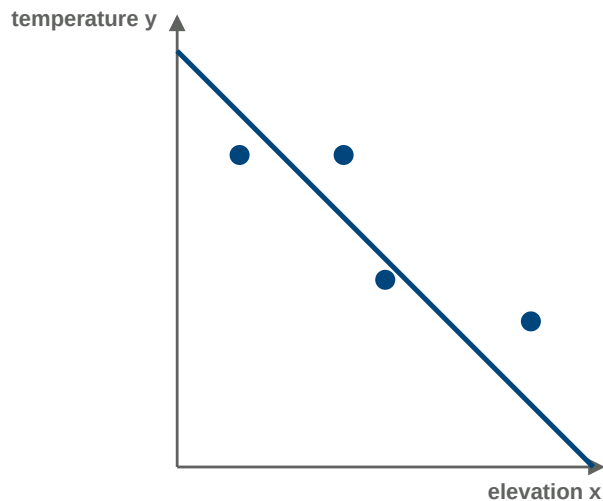
A simple (but powerful) linear model

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

maps an elevation x to an estimated temperature $\hat{y}$, with β_0 the y-intercept and β_1 the slope.

Figure: M. Rußwurm & D. Tuia, *Linear Regression*, ENV-408 course materials, EPFL, 2023.

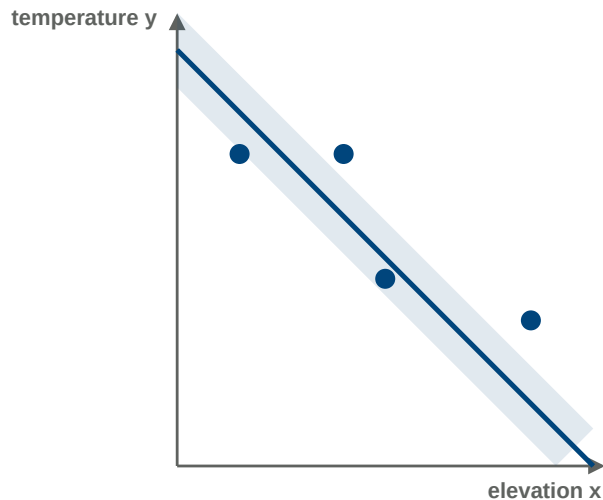
The Problem: Finding $\hat{\beta}$



We have a dataset of data samples (x_i, y_i) .

Problem: how do we find good model parameters
 $\hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1)$ for our linear model $f(x; \hat{\beta}) = \hat{\beta}_0 + x\hat{\beta}_1$?

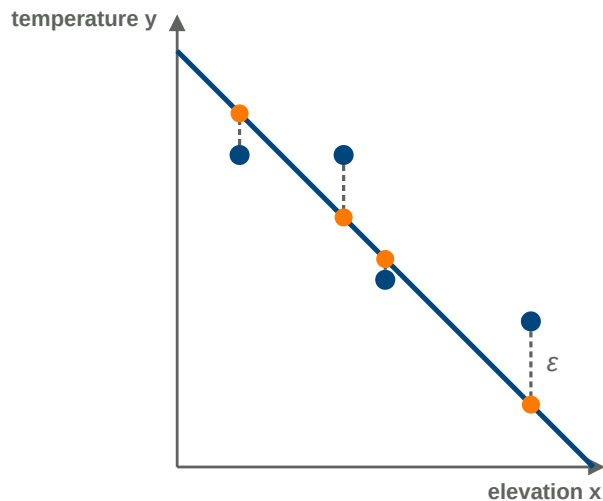
What Is a Good Model — Formally?



$$y_i = f(x_i; \hat{\beta}) \pm \overset{\text{error}}{\varepsilon}$$
$$\hat{y}_i = f(x_i; \hat{\beta})$$

with the error ε as a random variable — the model never fits the data perfectly.

Residuals



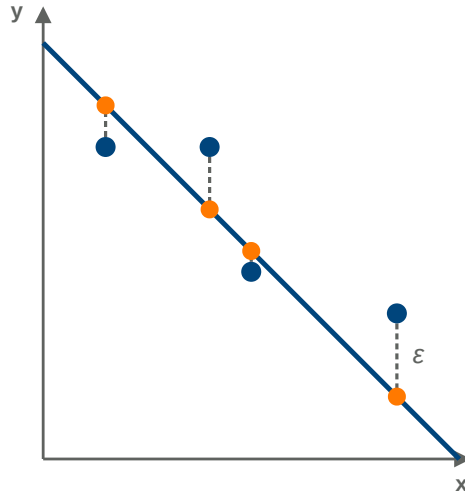
Concretely, the error — the "residual" ε_i — for each sample i :

$$|\hat{y}_i - y_i| = \varepsilon_i$$

We want to minimize the residuals over the whole dataset

$$\min \frac{1}{N} \sum_{i=1}^N \varepsilon_i$$

Mean Absolute Error (MAE)

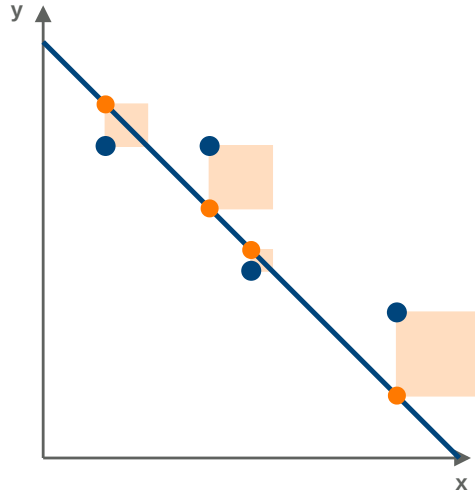


$$\text{MAE}(\mathcal{D}, \beta) = \frac{1}{N} \sum_{i=1}^N |\hat{y}_i - y_i|$$

over model β with $\hat{y}_i = f(x_i; \beta)$ and dataset $\mathcal{D} = \{x_i, y_i\}_{i=1}^N$

Very intuitive — but does not punish large errors.

Mean Squared Error (MSE)



$$\text{MSE}(\mathcal{D}, \beta) = \frac{1}{2N} \sum_{i=1}^N (\hat{y}_i - y_i)^2$$

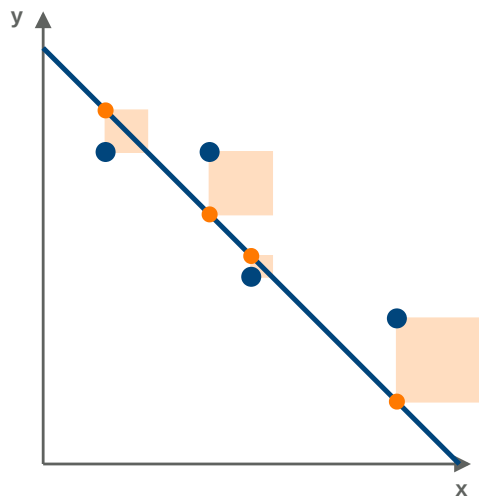
Punishes large errors with a square penalty.

Geometric intuition: find a model that minimizes the total area of these squares.

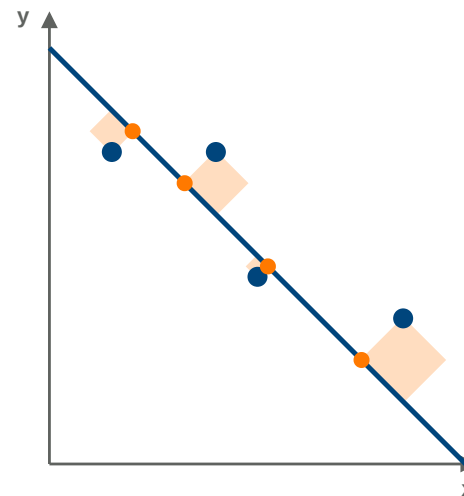
Vertical vs. Perpendicular Error Offsets

Both measure a residual — but they disagree on what "error" means for a spatial variable.

Vertical offsets



Perpendicular offsets



Ordinary Least Squares (OLS) — Intuition

The closed-form solution $\hat{\beta}_0, \hat{\beta}_1$ for a linear model under MSE.

Task: find $\hat{\beta}_0, \hat{\beta}_1$ that minimize the Mean Squared Error

$$\hat{\beta}_0, \hat{\beta}_1 = \arg \min_{\beta_0, \beta_1} \frac{1}{2N} \sum_{i=1}^N \left(\overbrace{\beta_0 + x_i \beta_1}^{\hat{y}_i} - y_i \right)^2$$

Question: how do we minimize a function with respect to its parameters?

- calculate the partial derivatives $\frac{\partial \text{MSE}}{\partial \beta_0}$ and $\frac{\partial \text{MSE}}{\partial \beta_1}$
- set the derivatives to zero: $\frac{\partial \text{MSE}}{\partial \beta_0} = 0$, $\frac{\partial \text{MSE}}{\partial \beta_1} = 0$
- solve for β_0 and β_1

Ordinary Least Squares (OLS) — Solution

Model $y = \beta_0 + x\beta_1$ on a dataset $\mathcal{D} = \{x_i, y_i\}_{i=1}^N$

Ordinary Least Squares objective

$$\hat{\beta}_0, \hat{\beta}_1 = \arg \min_{\beta_0, \beta_1} \frac{1}{2N} \sum_{i=1}^N \left(\overbrace{\beta_0 + x_i \beta_1}^{\hat{y}_i} - y_i \right)^2$$

Closed-form solution

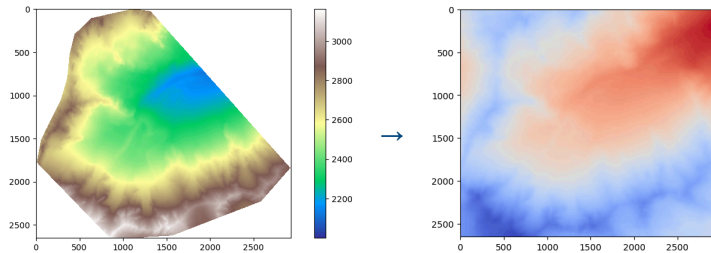
$$\begin{aligned} \hat{\beta}_0 &= \frac{1}{N} \sum_{i=1}^N y_i - \hat{\beta}_1 \frac{1}{N} \sum_{i=1}^N x_i = \bar{y} - \hat{\beta}_1 \bar{x} \\ \hat{\beta}_1 &= \frac{\text{Cov}[x, y]}{\text{Var}[x]} = \frac{\frac{1}{N} \sum_i (x_i - \bar{x})(y_i - \bar{y})}{\frac{1}{N} \sum_i (x_i - \bar{x})^2} \end{aligned}$$

See [Ordinary Least Squares](#), Wikipedia.

Limits of Univariate Models

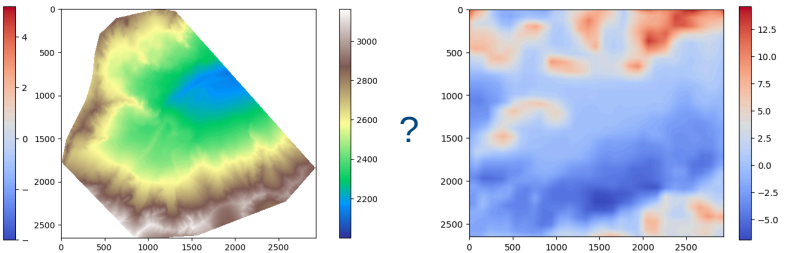
Some relationships are well-described by a single variable — most are not.

Roughly linear



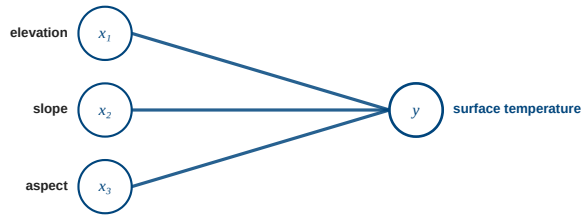
elevation $\rightarrow$ average temperature, roughly $-0.006 \frac{^{\circ}\text{C}}{\text{m}}$

Clearly non-linear



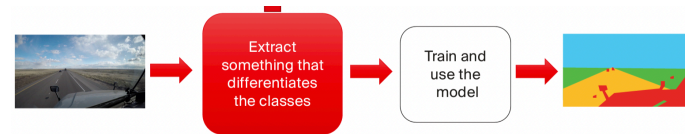
elevation $\rightarrow$ surface temperature — shading, wind, and local terrain also matter

Multivariate Models: Linearize With Good Feature Design



Many non-linear relationships can be linearized with good feature design: elevation alone under-explains surface temperature, but elevation **and** slope **and** aspect together explain it much better.

Same pattern as classical feature extraction: extract informative features, then fit a simple model.



From Uni- to Multivariate Linear Models

Univariate model $y = f(x; \beta_0, \beta_1)$ with scalar x, y and two parameters

$$y = \beta_0 + \overbrace{x_1}^{\text{e.g., elev.}} \beta_1$$

Multivariate linear model

- one input sample $\mathbf{x}_i = (1, x_1, x_2, x_3, \dots, x_p)$ of p features
- corresponding target y_i
- one weight per feature $\boldsymbol{\beta} = (\beta_0, \beta_1, \beta_2, \dots, \beta_p)$

$$y_i = \beta_0 + \overbrace{x_1}^{\text{e.g., elev.}} \beta_1 + \overbrace{x_2}^{\text{e.g., slope}} \beta_2 + \overbrace{x_3}^{\text{e.g., aspect}} \beta_3 + \dots = \boldsymbol{\beta}^\top \mathbf{x}_i$$

Multivariate Linear Model in Matrix Form

A multivariate dataset $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^n$ as matrix $\mathbf{X}$ and target vector $\mathbf{y}$, with n samples and p features

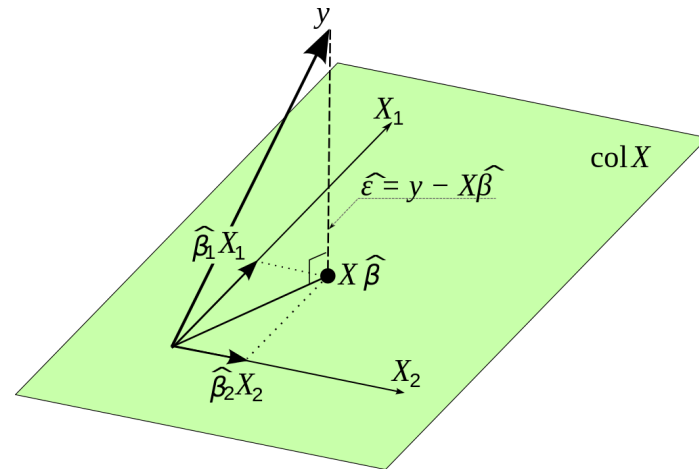
$$\mathbf{X} = \begin{pmatrix} 1 & \widehat{\text{elev.}} & \widehat{\text{slope}} & \widehat{\text{aspect}} & \dots \\ 1 & x_{11} & x_{21} & x_{31} & \dots \\ 1 & x_{12} & x_{22} & x_{32} & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \\ 1 & x_{1n} & x_{2n} & x_{3n} & \dots \end{pmatrix}, \quad \mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

And the model

$$\mathbf{y} = \mathbf{X}\hat{\boldsymbol{\beta}} + \widehat{\boldsymbol{\varepsilon}}^{\text{errors}}$$

with $p+1$ parameters $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_p)$

$$\boldsymbol{\varepsilon} = \overbrace{\|\mathbf{X}\hat{\boldsymbol{\beta}} - \mathbf{y}\|}^{\text{MSE}}^2$$



Multivariate Ordinary Least Squares

$$\hat{\beta} = \arg \min_{\hat{\beta}} \text{MSE}(\hat{\beta}) = \arg \min_{\hat{\beta}} \left\| \mathbf{X}\hat{\beta} - \mathbf{y} \right\|^2$$

Question: how do we minimize a function with respect to its parameters — now a whole vector $\hat{\beta}$?

- calculate the partial derivatives $\frac{\partial \text{MSE}}{\partial \hat{\beta}}$
- set the derivatives to zero: $\frac{\partial \text{MSE}}{\partial \hat{\beta}} = 0$
- solve for $\hat{\beta}$

Multivariate OLS — Closed-form Derivation

With Mean Squared Error

$$\text{MSE}(\hat{\beta}) = \left\| \mathbf{y} - \mathbf{X}\hat{\beta} \right\|^2 = \mathbf{y}^\top \mathbf{y} - 2\hat{\beta}^\top \mathbf{X}^\top \mathbf{y} + \hat{\beta}^\top \mathbf{X}^\top \mathbf{X} \hat{\beta}$$

and its derivative set to zero

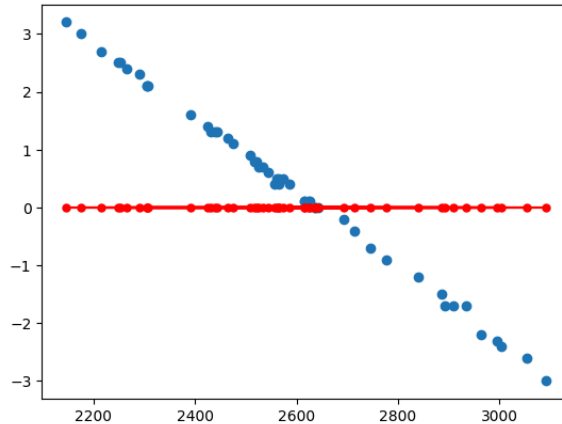
$$\frac{\partial \text{MSE}(\hat{\beta})}{\partial \hat{\beta}} = -2\mathbf{X}^\top \mathbf{y} + 2\mathbf{X}^\top \mathbf{X} \hat{\beta} = 0$$

$$\mathbf{X}^\top \mathbf{X} \hat{\beta} = \mathbf{X}^\top \mathbf{y}$$

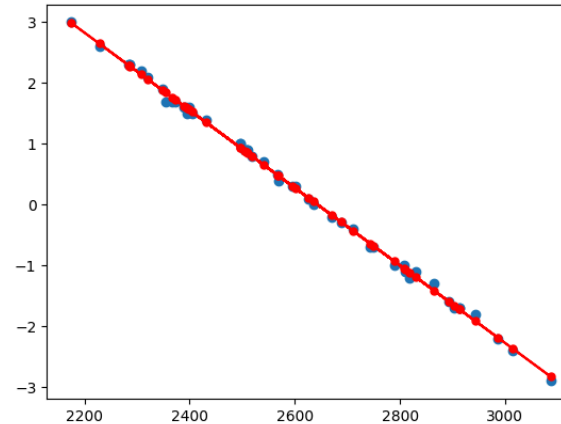
$$\hat{\beta} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$$

This is the normal equation — a closed-form solution, no iterative optimization required.

Let's Appreciate the Elegance



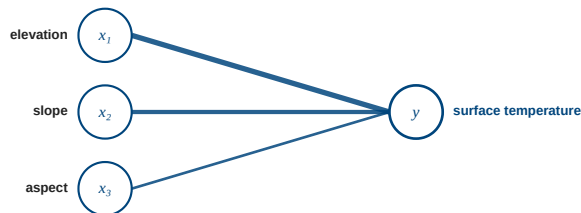
$$\beta = (0, 0, 0, 0, \dots)$$



$$\beta = (-.1, .2, .5, -.5, \dots)$$

$\hat{\beta} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$ — a single line of code, and blazing fast.

Interpretable Coefficients

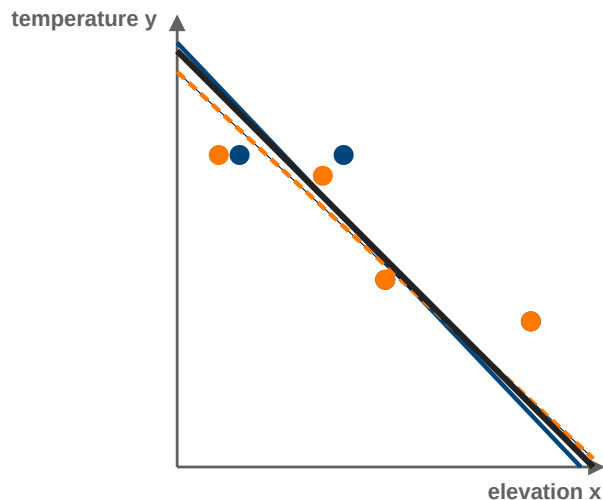


```
coefficients
beta 0 y-intercept : 9.769941
beta 1 dem          : -0.004522
beta 2 slope        : 0.032874
beta 3 dem_hor      : -0.026361
beta 4 dem_vert     : 0.222707
beta 5 aspect       : 0.002971
beta 6 diff_dem     : 0.012950
```

Each $\hat{\beta}_j$ tells us how $\hat{y}$ changes per unit of x_j , holding the other features fixed.

Caveat: coefficients are only directly comparable across features when the features share units — or are normalized.

Model Evaluation



We can now fit a model $(\hat{\beta}_0, \hat{\beta}_1, \dots)$ to any set of data samples $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ — but different samples give different fits.

$$Y = \beta_0 + \beta_1 X_1 + \dots + \varepsilon$$

Did we capture the true, underlying relationship $(\beta_0, \beta_1, \dots)$?

1. how large is the data error ε ? → RSE
2. does our model explain the target variable well? → R^2
3. how close is $\hat{\beta}$ to the true β ? → $SE[\hat{\beta}]$

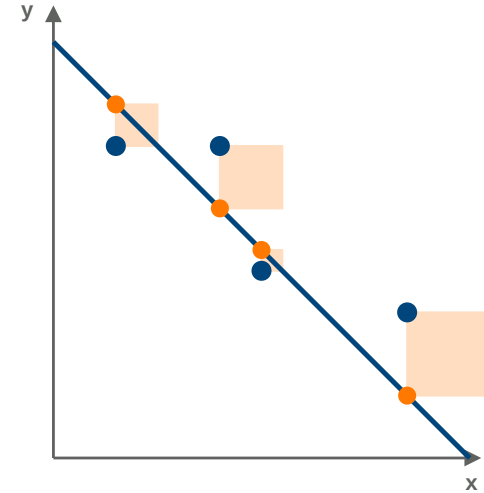
→ a hypothesis test tells us whether the relationship is significant given the dataset size.

Residual Sum of Squares (RSS)

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

RSS increases with the dataset size N .

Idea: normalize by the number of data samples.



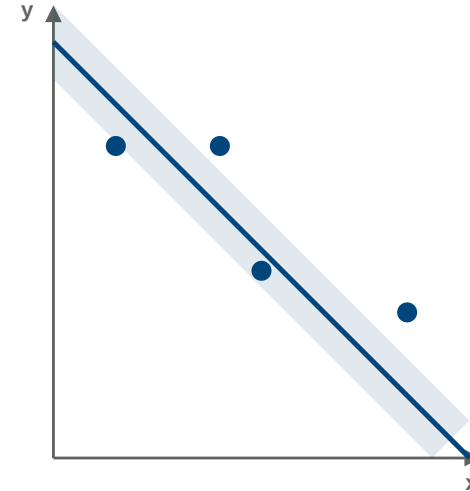
Residual Standard Error (RSE)

$$\text{RSE} = \sqrt{\underbrace{\frac{1}{N - M - 1}}_{\text{DoF}}} \text{RSS}$$

- normalized by the number of samples N
- and the degrees of freedom (DoF): M slope parameters and one y-intercept

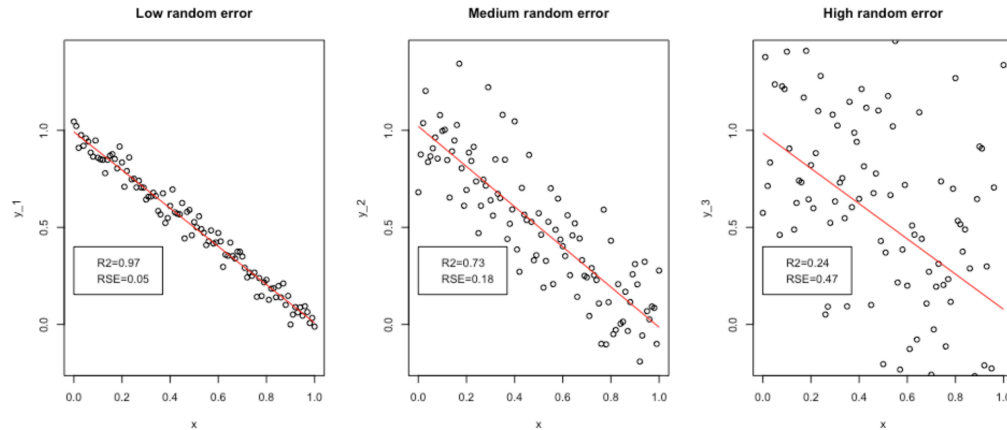
The RSE estimates $\text{std}(\varepsilon)$ and is a good measure of fit.

→ but it is in the units of the target variable: what number counts as a good fit?



R² Metric: Coefficient of Determination

The R² is a **unitless proportion**, typically between 0 and 1: it measures how much of the variability in Y is explained by $\hat{Y}$.



In the univariate case, R² is the square of the **Pearson correlation coefficient** $r(Y, \hat{Y}) = \frac{\text{Cov}[Y, \hat{Y}]}{\sigma_Y \sigma_{\hat{Y}}}$.

Figure: M. Rußwurm & D. Tuia, *Linear Regression*, ENV-408 course materials, EPFL, 2023. See also [Coefficient of Determination](#), Wikipedia.

R²: Comparison to an Averaging Model

How much better is our model than a simple averaging model that always predicts $\bar{y}$, no matter the input x ?

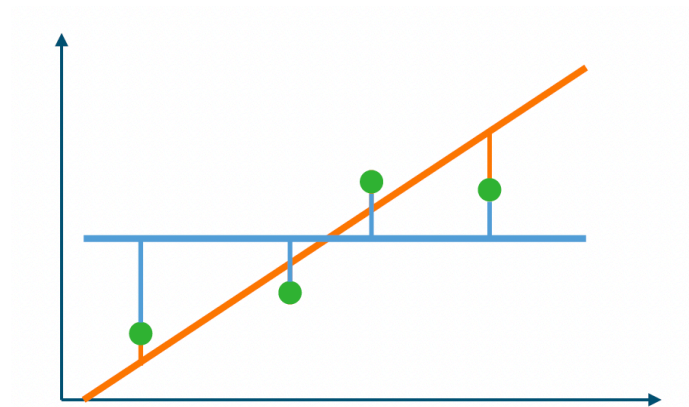
$$R^2 = 1 - \frac{\text{RSS}}{\text{TSS}}$$

Residual Sum of Squares

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

Total Sum of Squares

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



Estimating the Standard Error for Coefficients

In the underlying model

$$Y = \beta_0 + \beta_1 X_1 + \dots + \varepsilon$$

we can estimate the residual standard error $\sigma = \text{std}(\varepsilon) = \text{RSE}$ — the better the fit, the smaller the standard deviation of ε .

How close are the estimated coefficients $\hat{\beta}$ to the true, underlying relationship β ?

We can calculate the Standard Error (SE) for each coefficient:

$$\text{SE}[\hat{\beta}_0] = \sigma^2 \left[\frac{1}{N} + \frac{\bar{x}^2}{\sum_{i=1}^N (x_i - \bar{x})^2} \right], \quad \text{SE}[\hat{\beta}_1] = \frac{\sigma^2}{\sum_{i=1}^N (x_i - \bar{x})^2}$$

Student's t-test: Is Our Model Significant?

Question: is an estimated relationship a random artifact of the data?

Model: $Y = \beta_0 + \beta_1 X_1 + \varepsilon$

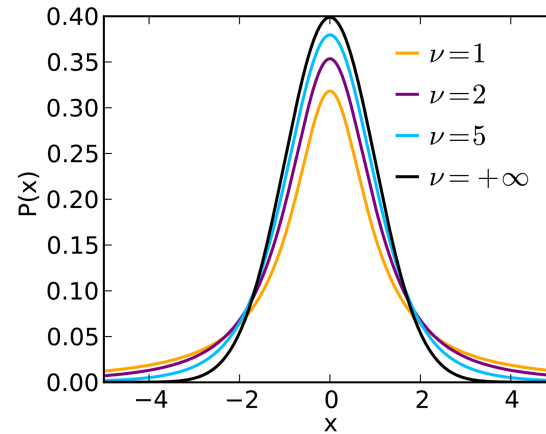
Testing the slope β_1

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

Test statistic

$$t_0 = \frac{\hat{\beta}_1}{\text{SE}[\hat{\beta}_1]}$$

Reject H_0 if $t_0 < -t_{\alpha, N-2}$ or $t_0 > t_{\alpha, N-2}$



$t_{\alpha, N-2}$ comes from the Student-t distribution with $\nu = N - 2$ degrees of freedom.

Figure: Wikipedia, author Skbkakas, [Student's t-distribution](#). CC BY-SA.

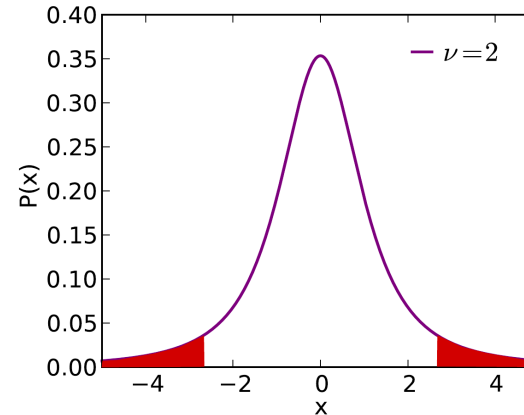
P-value: The Probability of a Lucky Shot

The (two-sided) p-value integrates the Student-t distribution to t_0 on both sides.

high p-value → not enough data

high p-value → no relationship

We can use the p-value of different coefficients to select significant features.



p-value shown in red

Figure modified from Wikipedia, original author Skbkekas, [Student's t-distribution](#), CC BY-SA.

Recap: Linear Regression

This part covered how to:

- implement a univariate and multivariate linear model from scratch
- fit the parameters with the Ordinary Least Squares (OLS) method
- evaluate the goodness of fit with RSE and R^2
- test individual parameters for significance with Student's t-test
- select features dynamically based on their significance

Next: logistic regression extends the same linear model to map a fixed representation to categories with a linear decision boundary.

Non-linear models: Decision Trees & Random Forests



Image by Bing Image Generator

