AI4 – Machine Learning

Linear Regression

Goals:

- You know the difference between classification and regression.

- You are able to apply (linear) regression on a dataset.

Fontys

DENK GROTER

Types of Numerical Data

- Discrete
  - Countable set of values (zip-codes, # good, # red, etc.)
  - Commonly represented by integer variables

- Continuous
  - Ratios (% interest)
  - Physical distances (for example how tall a building is)
  - Temperature
  - Floating point variabel (or is it?)

Are Continuous Variables Really Continuous? 🤔

We often think of things like **temperature, speed, and voltage** as continuous. But in reality:

✔️ **Computers store numbers with limited precision** (floating-point numbers have a fixed number of bits).

✔️ **Measurements always have limits** (a thermometer can't measure temperature to infinite decimal places).

🔹 Most of the time, this doesn’t matter.

🔹 But in some cases (like scientific computing or machine learning), rounding errors can cause problems!

It is **Context Dependent.**

Know it and use it

Problem	Algorithm
Is this A or B?	Classification Algorithms
Is this weird?	Anomaly Detection Algorithms
How much? How many?	Regression Algorithms
How is this organized?	Clustering Algorithms
What should I do now?	Reinforcement Learning Algorithms

Examples of Everyday Regression & Prediction

- **Music & Recommendation Algorithms**
  “Spotify or Netflix predicts what you’ll like based on past choices.”

Examples of Everyday Regression & Prediction

- **Sports & Performance**
  “A coach predicts a player's future performance based on past training hours and fitness levels.”

Examples of Everyday Regression & Prediction

- **Commuting Time Prediction**
  “Your brain does regression daily: estimating the time it takes to get home based on past trips.”

Examples of Everyday Regression & Prediction

- **Social Interactions**
  “When texting a friend, you predict how long they’ll take to reply based on past behavior.”

Fontys

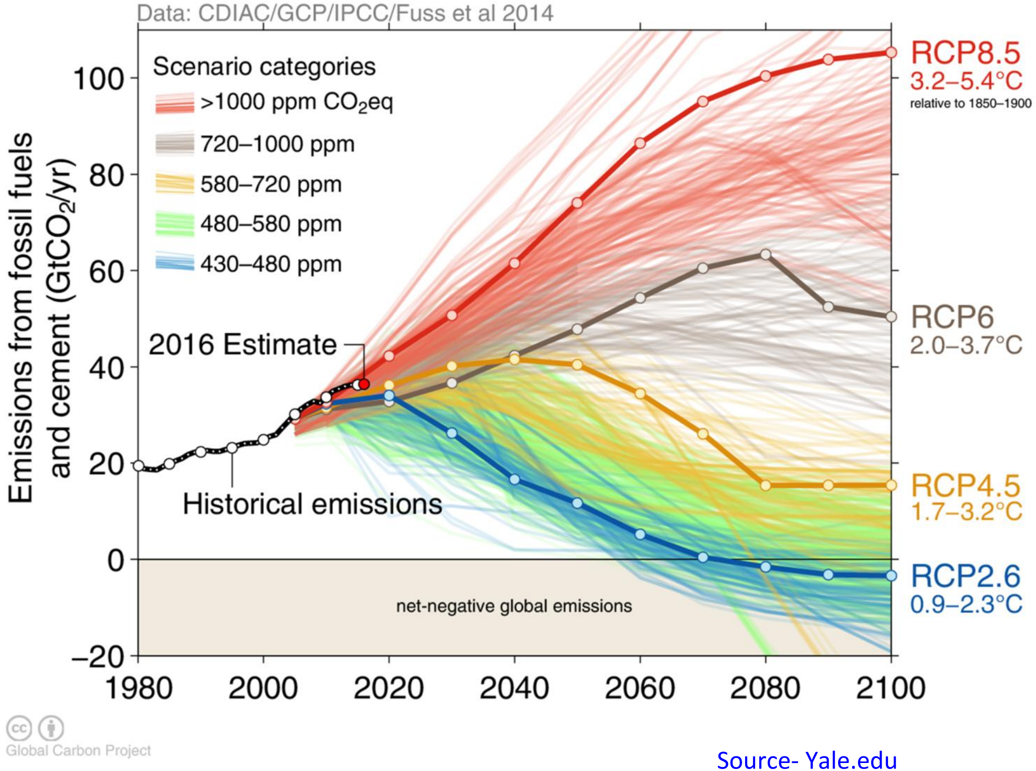
DENK GROTER

Your Turn

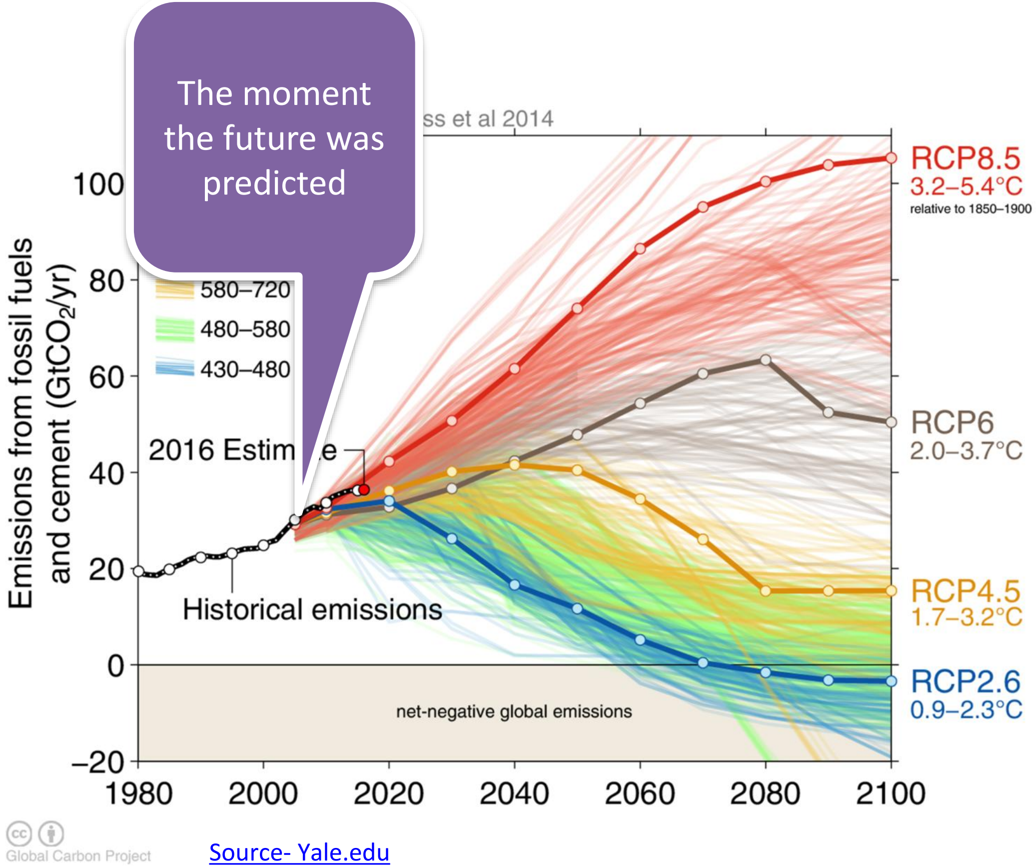
What was the last time you predicted something?

Did you ever made a critical prediction?

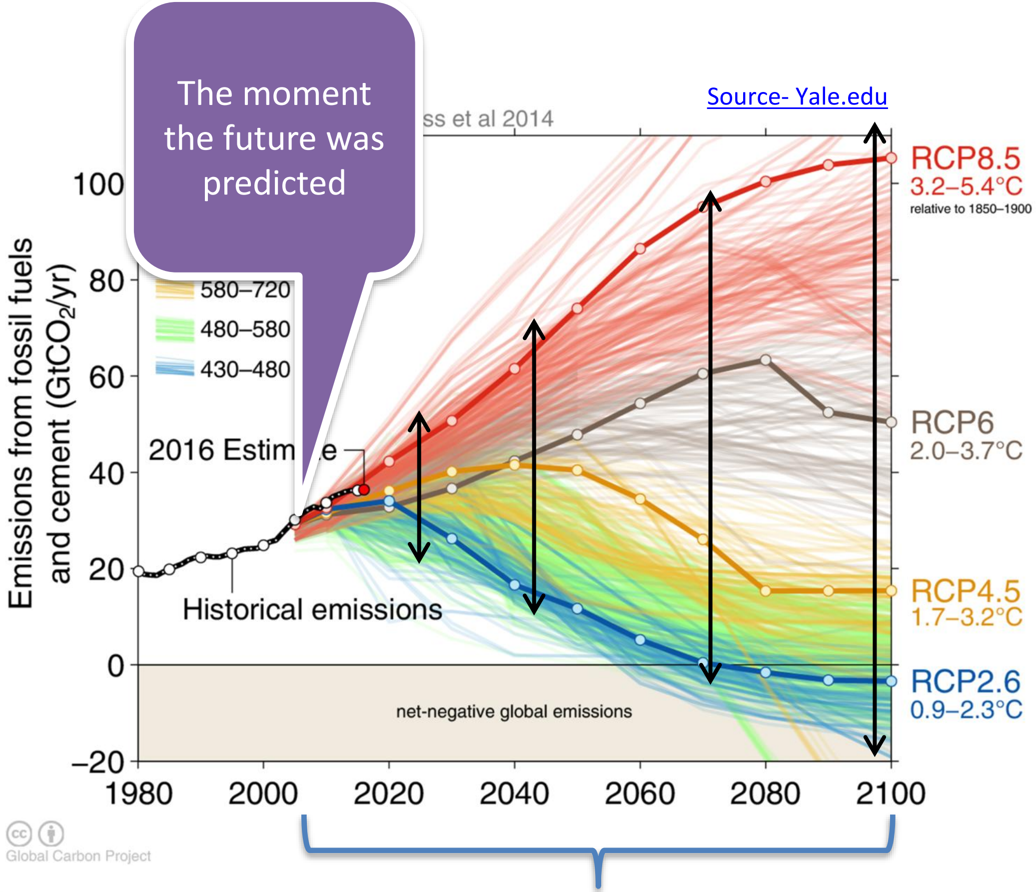
Example of Time Series Prediction



Source- Yale.edu



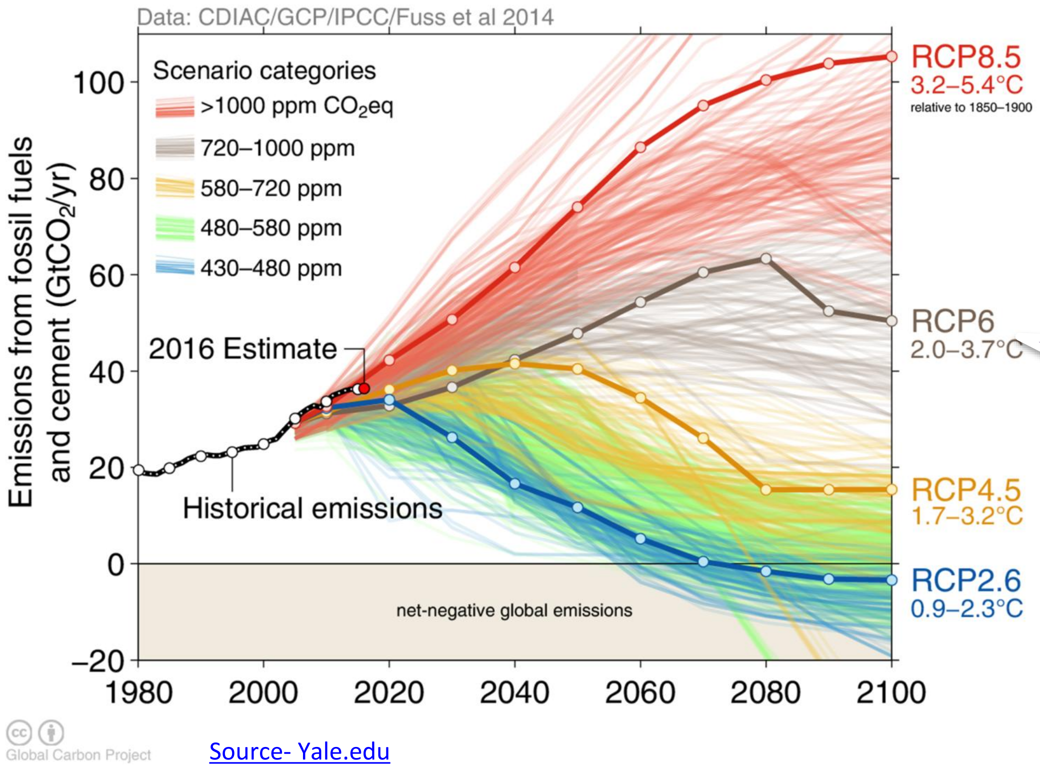
Source- Yale.edu



which trajectory will happen? As you look farther it gets less certain!

Fontys

DENK GROTER



Different colors depend on how much CO_2 will be emitted (different scenarios, policies, agreements, population size, technology advancements…)

Source- Yale.edu

Fontys

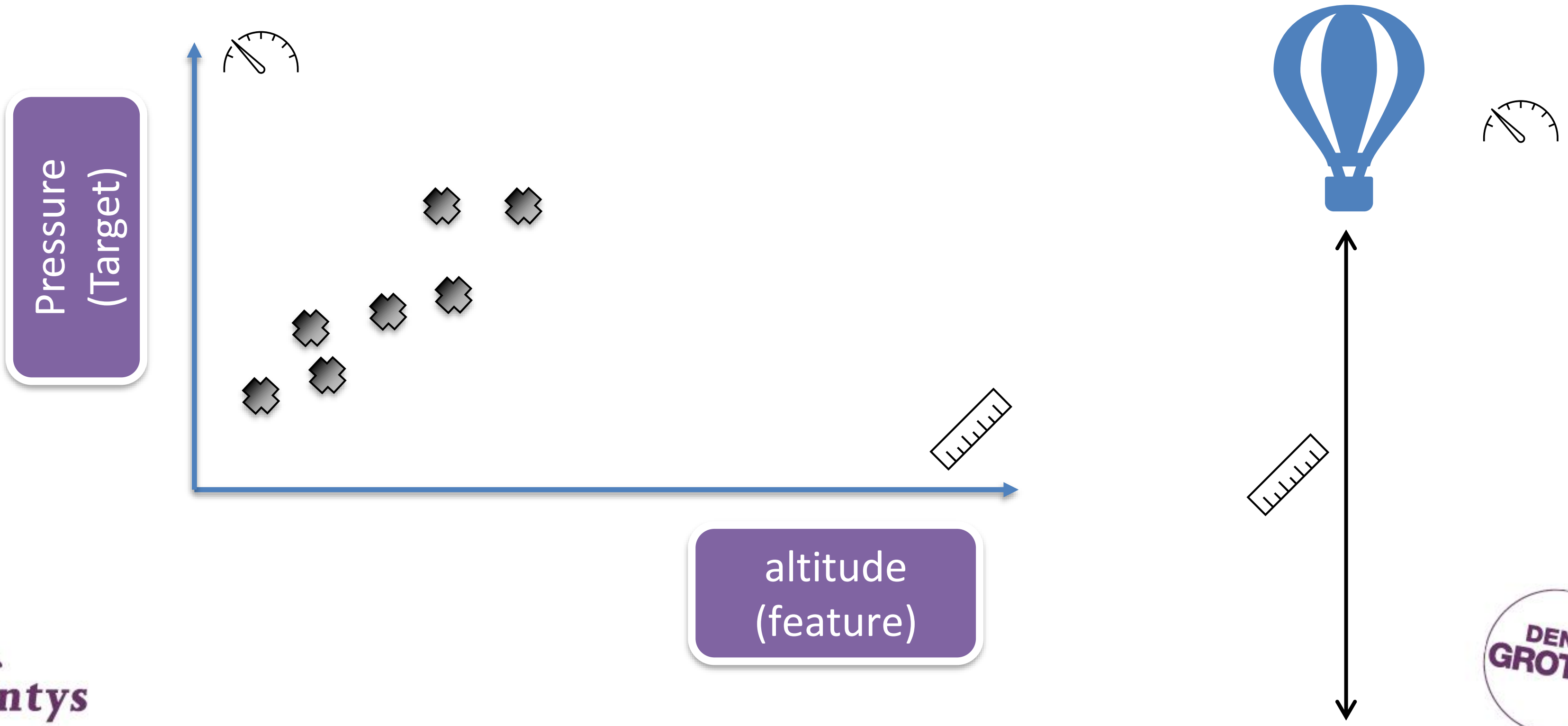
DENK GROTER

But how to make predictions?

- Find informative data
- Find a realistic model

- In this course we focus on one of the simplest models: The LINEAR model

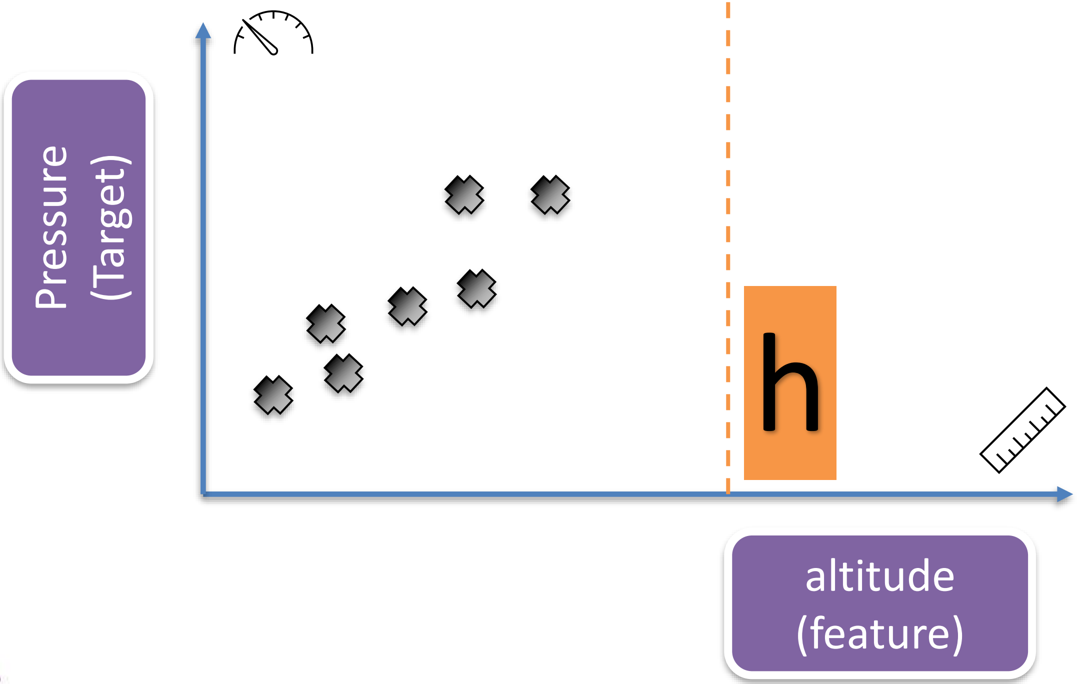
Let us measure the air pressure as we change our altitude

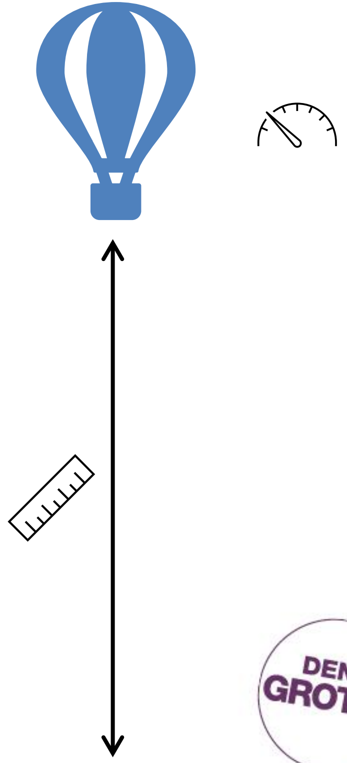


Fontys

DENK GROTER

Now we want to predict what the pressure is at a point we did not measure

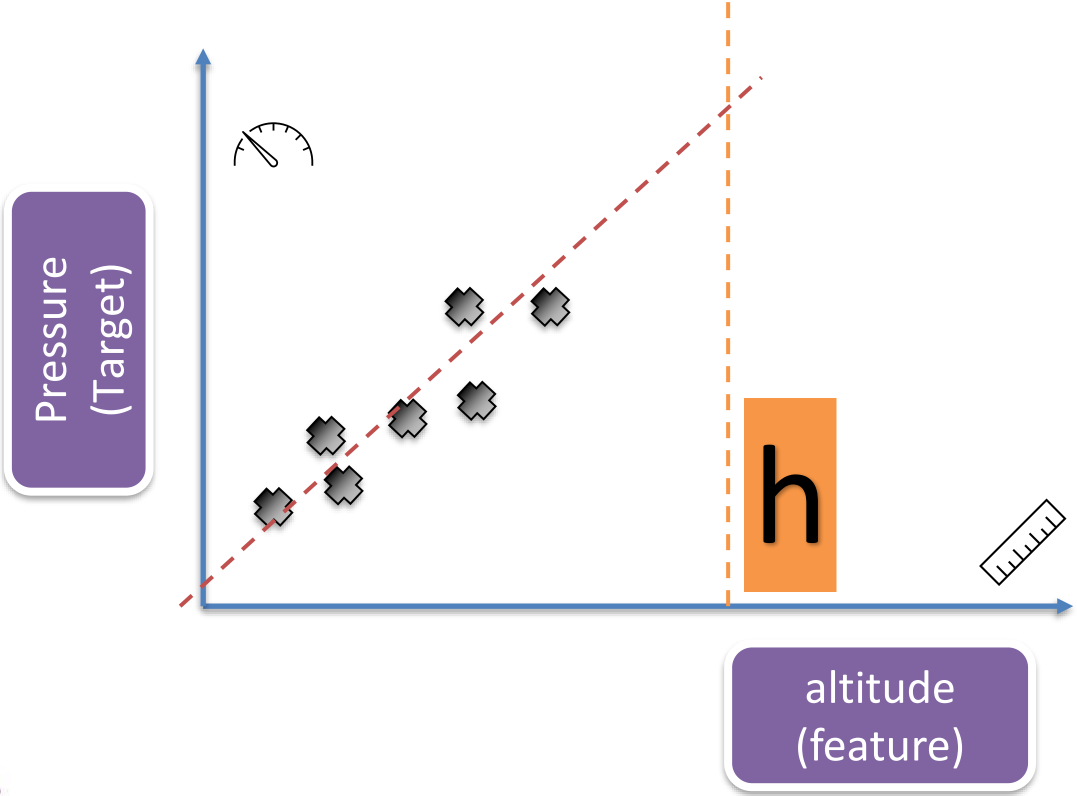


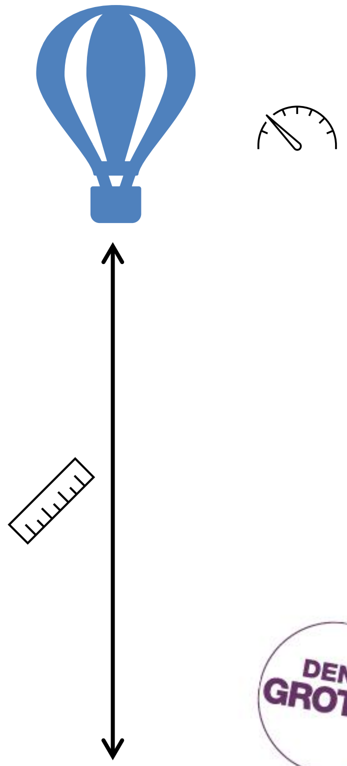


Fontys

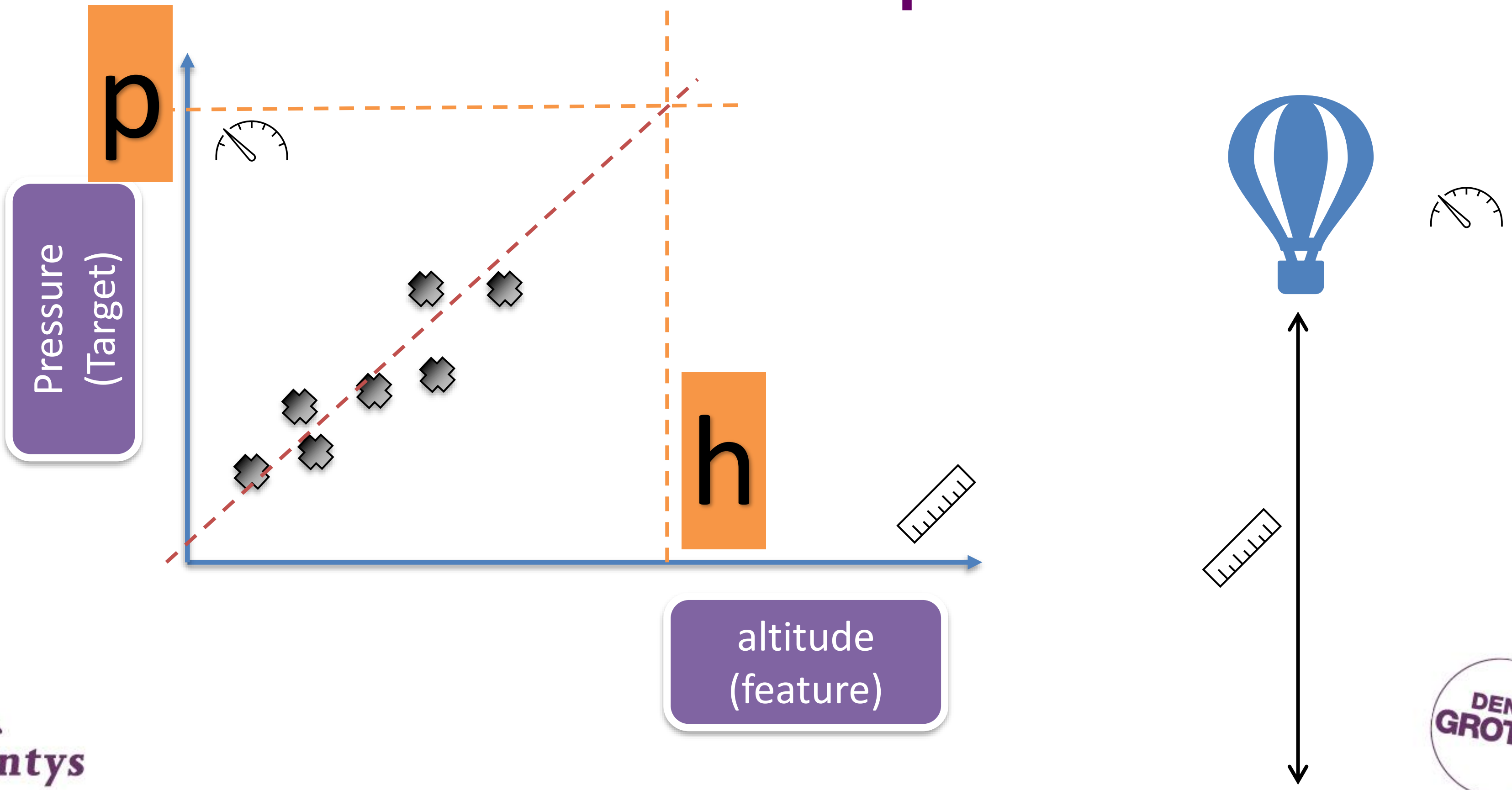
DENK GROTER

We model our data (here as a linear relationship)

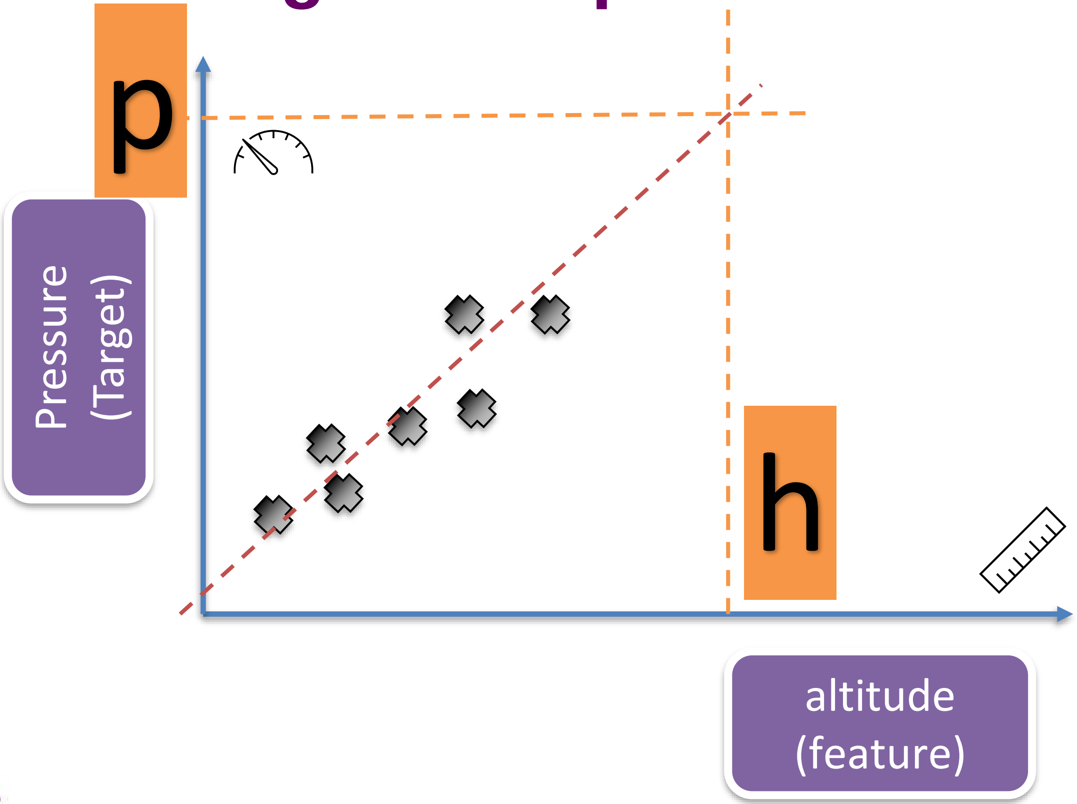


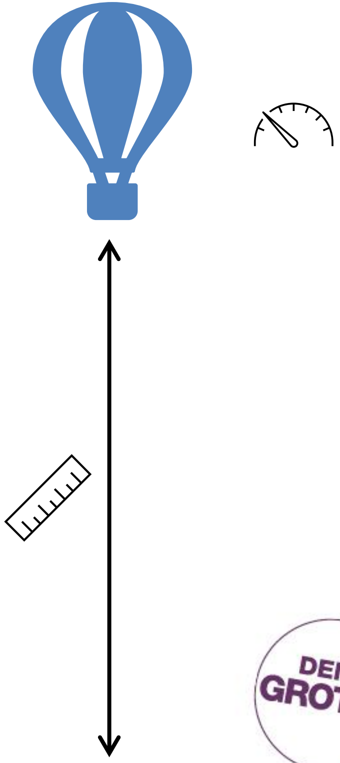


We use the model to make a prediction

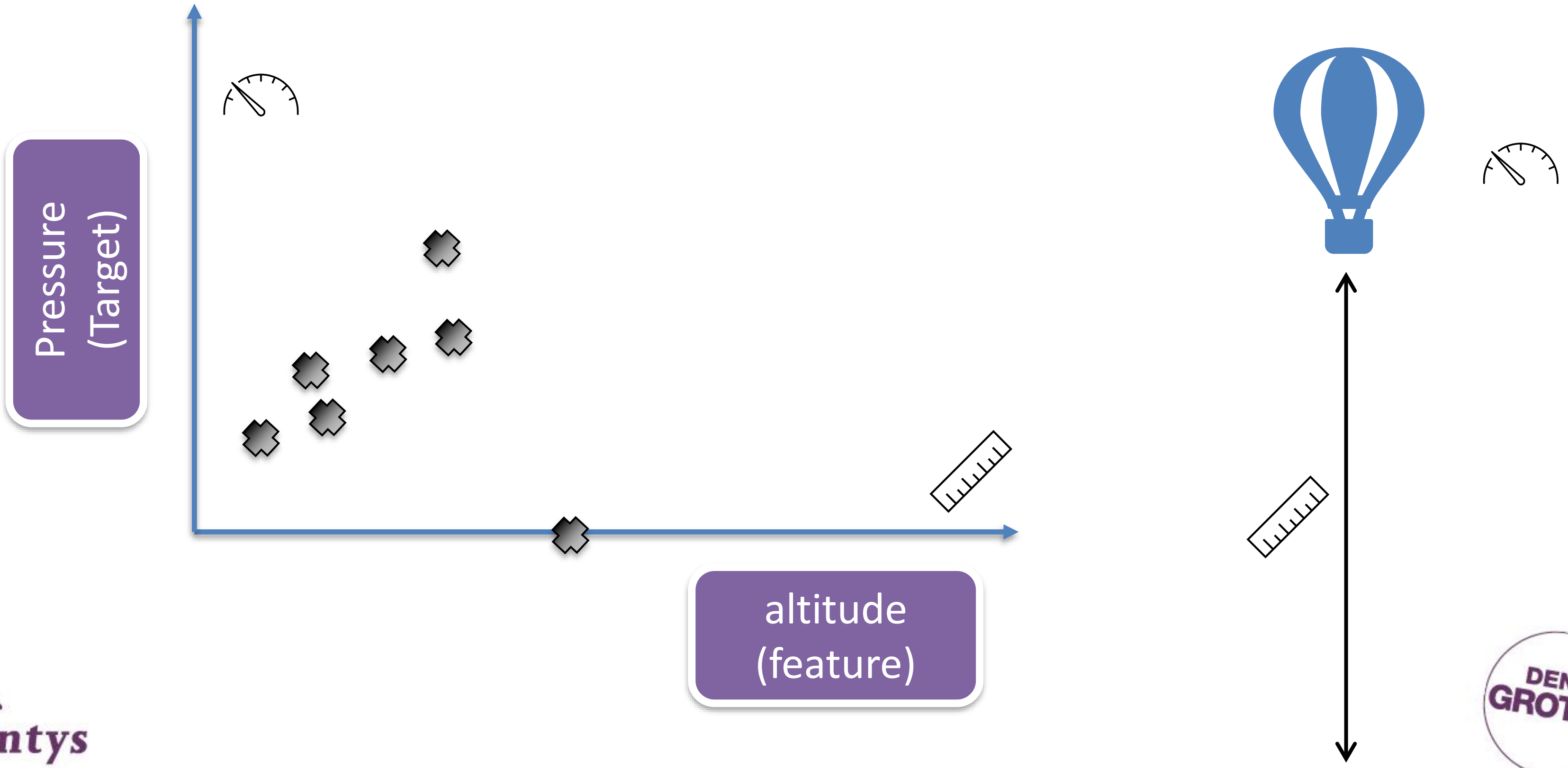


Predicting the air pressure

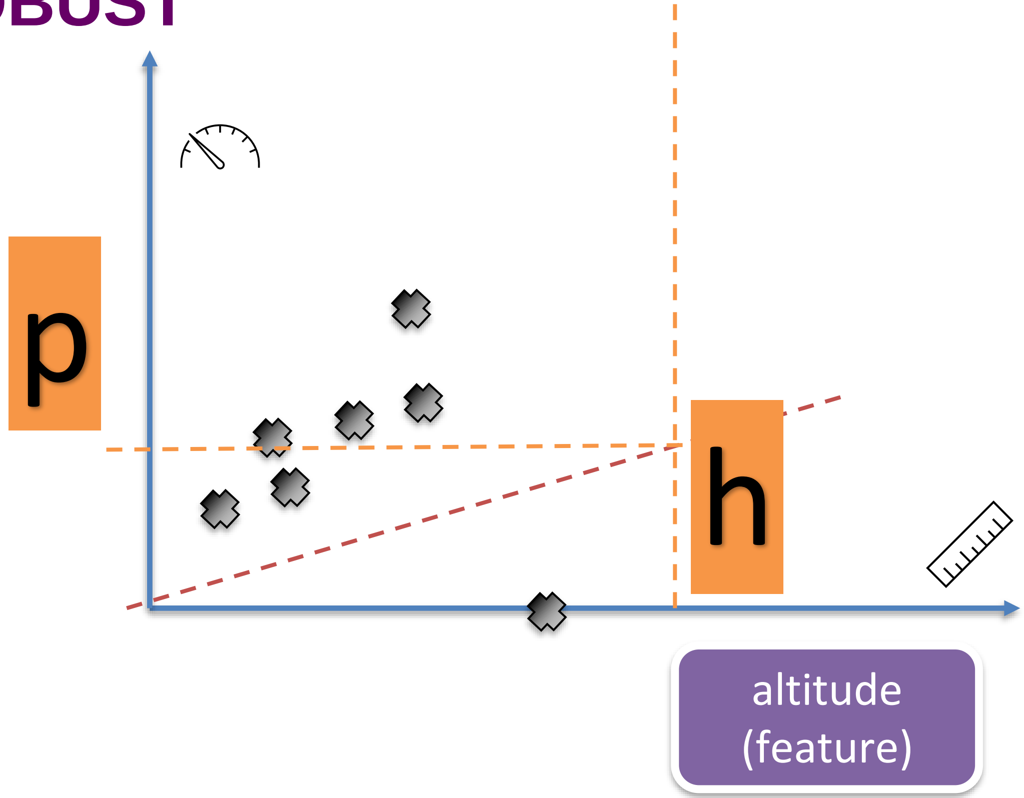


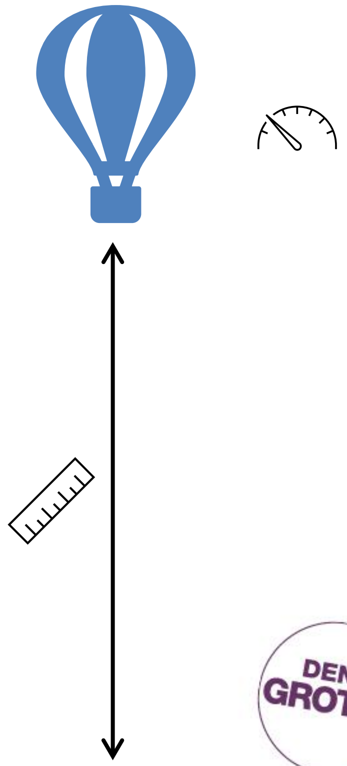


Outliers can be particularly harmful



Outliers can be harmful: Linear regression is not ROBUST



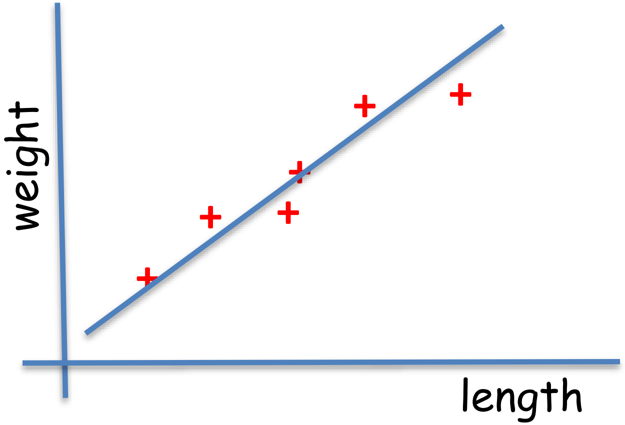


Fontys

DENK GROTER

Regression ≈ Continuous Labelling

- Create a (linear) relationship between 2 or more variables
- This enables you to predict something.
- Regression usually deals with continuous variables.
- Target: $y = f(x_1, x_2, \ldots)$
- Linear function is special case: $y = a \cdot x + b$

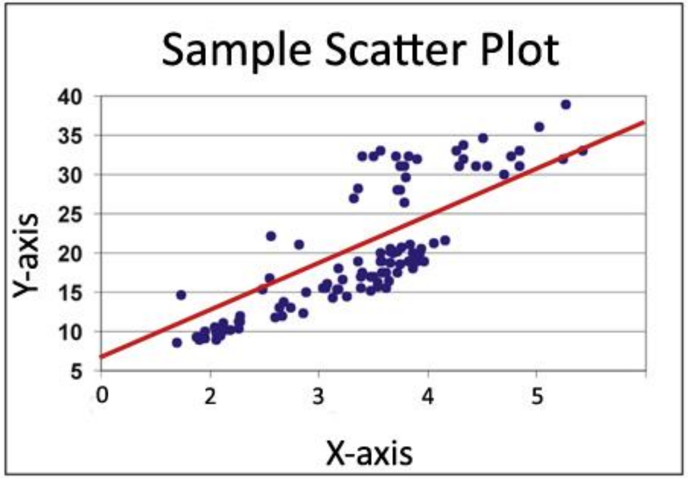


- *What is the difference with a scatter plot?*

Fontys

DENK GROTER

example: is data on a straight line? (y = a x + b)



x (axis): independent variable (feature)
y (axis): dependent variable (target)

We want the best estimate for a and b in
y = a x + b

Different Forms of Regression

- Linear regression:
  $y = b_0 + b_1 x$

- Multiple regression (multiple predictors):
  $y = b_0 + b_1 x_1 + b_2 x_2 + b_3 x_3 + \ldots$

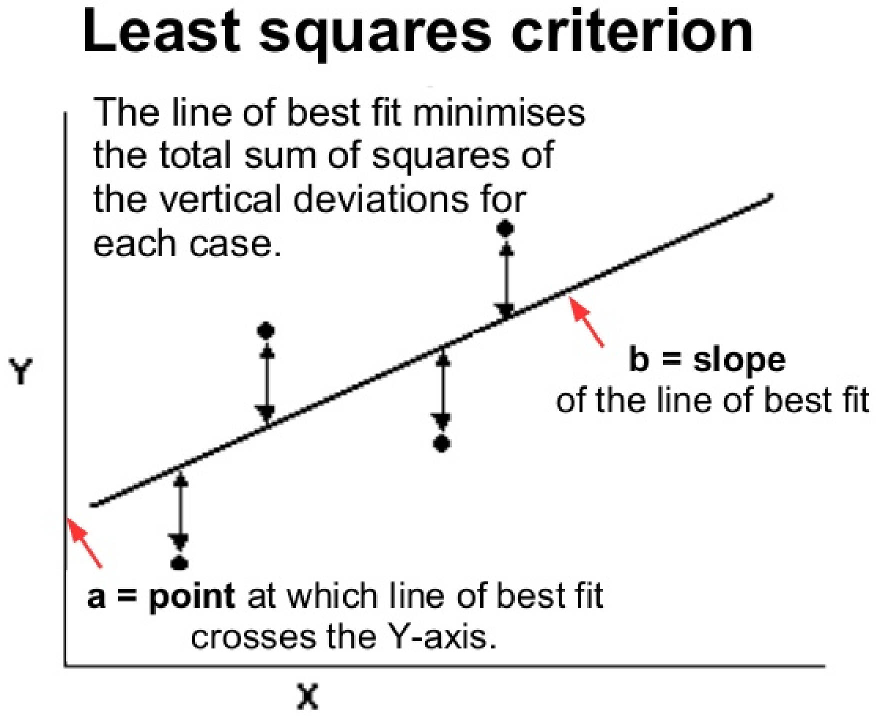
- Polynomial regression (non-linear in x but linear in coefficients b_0 , b_1, …):
  $y = b_0 + b_1 x^1 + b_2 x^2 + b_3 x^3 + \ldots$

Find and Evaluate Regression

- What is the best regression?
  - Find the curve
    (= parameter values for a, b, …)
    for which the
    Sum of Squared Errors (SSE)
    is minimal
  - This is a math problem
- How good is a regression?
  - SSE? *
  - R^2 value (between 0 and 1)

* SSE is not very reliable, it depends on the number of data points

But what does "best" mean?



The regression algorithm (e.g. the sklearn regressor) (calculates) the line with minimal sum of squared errors.

In other words: it minimizes the cost function that calculates the sum of squared errors for any pair of a (intercept) and b (slope).

R-squared

R-squared is a statistical measure (evaluation metric) of how close the data are to the fitted regression line. It solves some of the shortcomings of the sum-of-squares metric such as the dependence of number of training points.

R-squared is always between 0 and 100%:

- 0% indicates that the model explains none of the variability of the response data around its mean.
- 100% indicates that the model explains all the variability of the response data around its mean.

In general, the higher the R-squared, the better the model fits your data.

… but there is a lot of nuance to this!

Coding Linear Regression

- ```
  from sklearn.linear_model
    import LinearRegression              # include Linear Regression code
  ```

- ```
  reg = LinearRegression()               # Create regression object
  ```

- ```
  reg.fit(X_train, y_train)              # Fit regression line on
                                              train data
  ```

- ```
  pred = reg.predict(X_test)             # Predict Y values of testset
  ```

- ```
  r_square = reg.score(X_test, y_test) # Evaluate goodness-of-fit
  ```
  ```

Comparing Classification & Regression

Property	Supervised Classification	Regression
Output type (y)	Discrete (class labels)	Continuous (real number)
What are we trying to find?	Decision boundary	Best fit (curve)
Evaluation	Accuracy	r^2

There is a lot more to linear models

See also

`Ridge`
: Ridge regression addresses some of the problems of Ordinary Least Squares by imposing a penalty on the size of the coefficients with l2 regularization.

`Lasso`
: The Lasso is a linear model that estimates sparse coefficients with l1 regularization.

`ElasticNet`
: Elastic-Net is a linear regression model trained with both l1 and l2 -norm regularization of the coefficients.

They all fit a line, but **how** to fit it?

[LinearRegression — scikit-learn 1.6.1 documentation]()

There is a lot more to linear models

Ridge regression forces the model to **keep things smaller and neater**, so it doesn't overreact to small data changes.

See also

`Ridge`
: Ridge regression addresses some of the problems of Ordinary Least Squares by imposing a penalty on the size of the coefficients with l2 regularization.

`Lasso`
: The Lasso is a linear model that estimates sparse coefficients with l1 regularization.

`ElasticNet`
: Elastic-Net is a linear regression model trained with both l1 and l2 -norm regularization of the coefficients.

They all fit a line, but **how** to fit it?

[LinearRegression — scikit-learn 1.6.1 documentation]()

Fontys

DENK GROTER

There is a lot more to linear models

Lasso goes even further: it **can remove useless features** by forcing some coefficients to **exactly zero**.

See also

`Ridge`

Ridge regression addresses some of the problems of Ordinary Least Squares by imposing a penalty on the size of the coefficients with l2 regularization.

`Lasso`

The Lasso is a linear model that estimates sparse coefficients with l1 regularization.

`ElasticNet`

Elastic-Net is a linear regression model trained with both l1 and l2 -norm regularization of the coefficients.

They all fit a line, but **how** to fit it?

LinearRegression — scikit-learn 1.6.1 documentation

There is a lot more to linear models

See also

`Ridge`

Ridge regression addresses some of the problems of Ordinary Least Squares by imposing a penalty on the size of the coefficients with l2 regularization.

`Lasso`

The Lasso is a linear model that estimates sparse coefficients with l1 regularization.

`ElasticNet`

Elastic-Net is a linear regression model trained with both l1 and l2 -norm regularization of the coefficients.

Elastic Net – The Best of Both Worlds.
Elastic Net is like **a mix of Ridge and Lasso**

They all fit a line, but **how** to fit it?

[LinearRegression — scikit-learn 1.6.1 documentation]()

DENK GROTER

Polynomial Fitting [Optional, Self-learning]

Thes examples show different aspects of modeling data, the effect of noise, data-size, model-complexity in a hands-on way. You do not have to write code, but to change parameters in the code and make conclusions.

[learning_linear_models.ipynb – Colab]

[polynomial_model.ipynb – Colab]