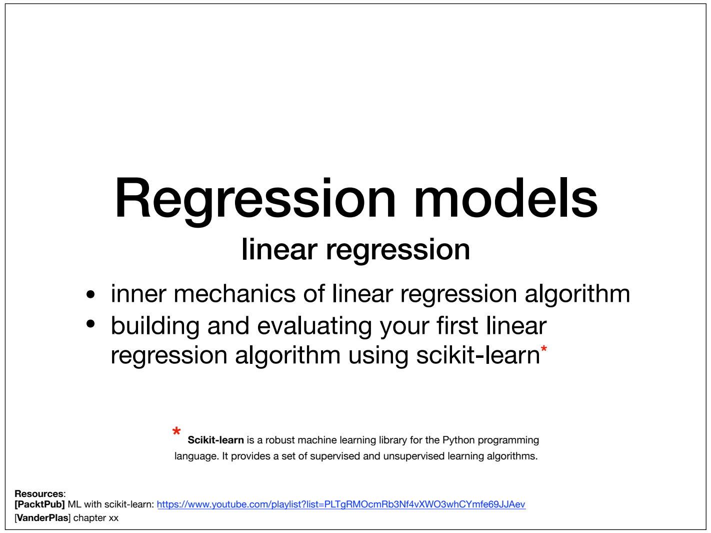


# Regression models

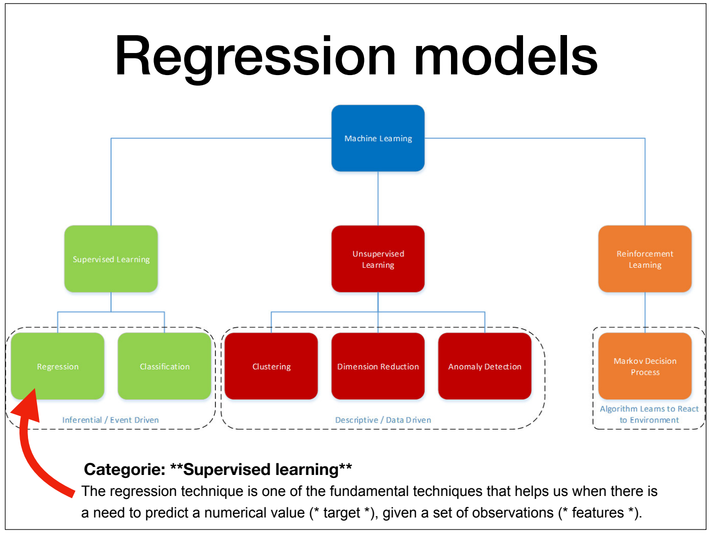
## linear regression

- inner mechanics of linear regression algorithm
- building and evaluating your first linear regression algorithm using scikit-learn*

* **Scikit-learn** is a robust machine learning library for the Python programming language. It provides a set of supervised and unsupervised learning algorithms.

**Resources**:
**[PacktPub]** ML with scikit-learn: https://www.youtube.com/playlist?list=PLTgRMOcmRb3Nf4vXWO3whCYmfe69JJAev
[**VanderPlas**] chapter xx

Scikit-learn is a robust machine learning library for the Python programming language. It provides a set of supervised and unsupervised learning algorithms.



# Regression models

Machine Learning

Supervised Learning

Unsupervised Learning

Reinforcement Learning

Regression

Classification

Clustering

Dimension Reduction

Anomaly Detection

Markov Decision Process

Inferential / Event Driven

Descriptive / Data Driven

Algorithm Learns to React to Environment

Categorie: **Supervised learning**

The regression technique is one of the fundamental techniques that helps us when there is a need to predict a numerical value (* target *), given a set of observations (* features *).

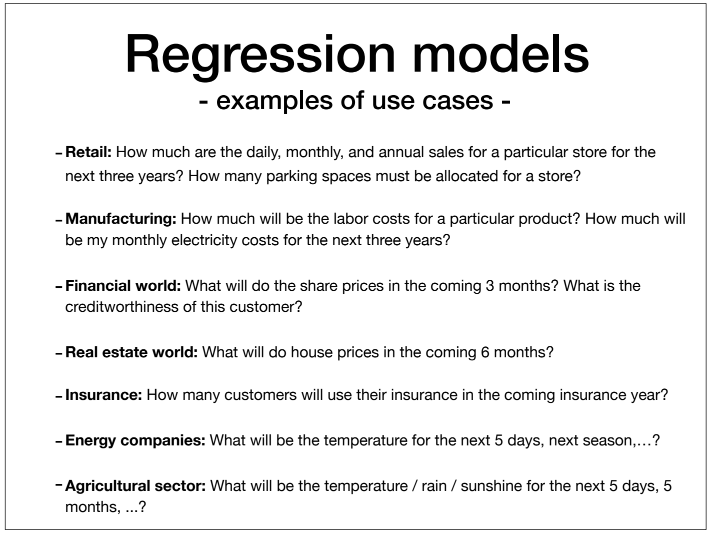
**Welke vragen/behoeften lost regression op?**
De regressie techniek is een van de fundamentele technieken die ons helpt wanneer er een behoefte is om een numerieke waarde (*target*) te voorspellen, gegeven een verzameling observaties (*features*).

Woorden in de vraagstelling/behoefte zoals '**hoeveelheid**', '**hoeveel keer**', verwachten een kwantitatief antwoord (*numerical, continuous*).

**Voorbeelden use cases**

- **Detailhandel**: Hoeveel zijn de dagelijkse, maandelijkse en jaarlijkse verkopen voor een bepaalde winkel voor de komende drie jaar? Hoeveel parkeerplaatsen moeten worden toegewezen voor een winkel?
- **Fabricage:** Hoeveel zullen de arbeidskosten zijn voor dit-en-dat product? Hoeveel zullen mijn maandelijkse elektriciteitskosten zijn voor de komende drie jaar?
- **Financieel wereld**: Wat zullen de aandelen prijzen doen de komende 3 maanden? Wat is de kredietwaardigheid van deze klant?

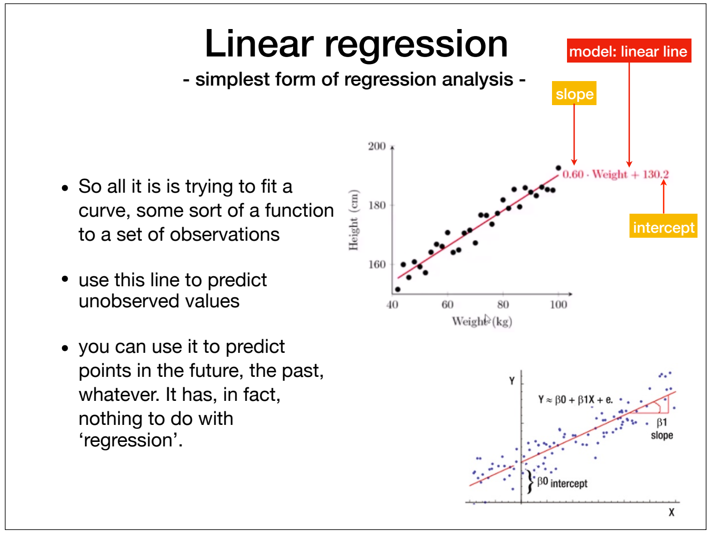
- **Onroerendgoed wereld**: Wat zullen de huizen prijzen doen de komende 6 maanden?
- **Verzekeringen**: Hoeveel klanten zullen een beroep doen op hun verzekeringen het komende verzekeringsjaar?
- **Energiebedrijven**: Wat zal de temperatuur zijn voor de volgende 5 dagen
- **Agrarische sector**: Wat zal de temperatuur/regen/zonneschijn zijn voor de volgende 5 dagen, 5 maanden,…?



# Regression models

## - examples of use cases -

- **Retail:** How much are the daily, monthly, and annual sales for a particular store for the next three years? How many parking spaces must be allocated for a store?
- **Manufacturing:** How much will be the labor costs for a particular product? How much will be my monthly electricity costs for the next three years?
- **Financial world:** What will do the share prices in the coming 3 months? What is the creditworthiness of this customer?
- **Real estate world:** What will do house prices in the coming 6 months?
- **Insurance:** How many customers will use their insurance in the coming insurance year?
- **Energy companies:** What will be the temperature for the next 5 days, next season,…?
- **Agricultural sector:** What will be the temperature / rain / sunshine for the next 5 days, 5 months, ...?



# Linear regression

- simplest form of regression analysis -

- So all it is is trying to fit a curve, some sort of a function to a set of observations
- use this line to predict unobserved values
- you can use it to predict points in the future, the past, whatever. It has, in fact, nothing to do with 'regression'.

model: linear line

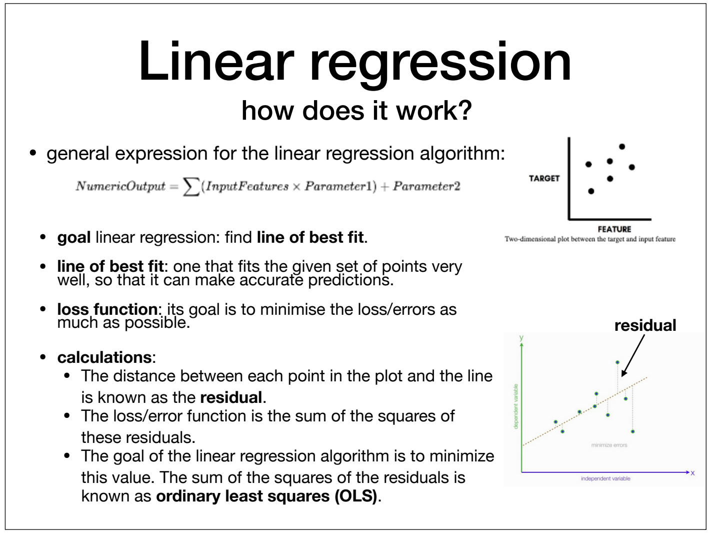
slope

intercept

lineaire regressie is alleen maar het passen van een rechte lijn voor een reeks observaties.

**prediction**

stel iemand weegt 120 kg, wat is zijn voorspelde lengte (height)? -> plm. 202 cm



# Linear regression

## how does it work?

- general expression for the linear regression algorithm:

$$NumericOutput = \sum(InputFeatures \times Parameter1) + Parameter2$$

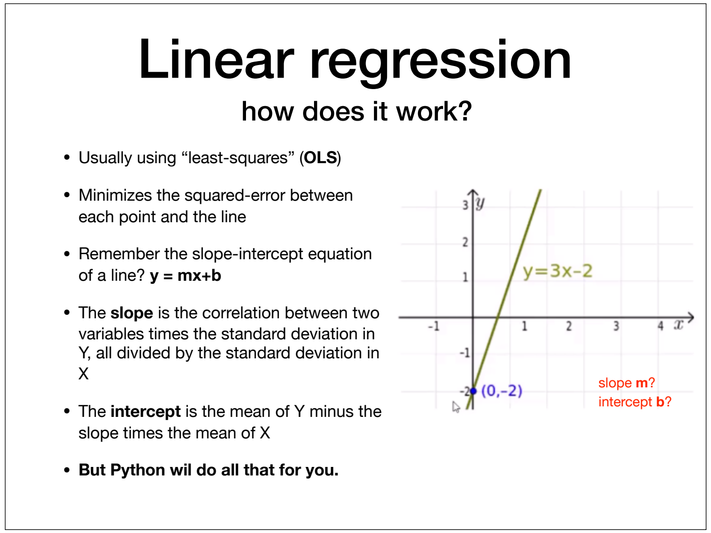
- **goal** linear regression: find **line of best fit**.
- **line of best fit**: one that fits the given set of points very well, so that it can make accurate predictions.
- **loss function**: its goal is to minimise the loss/errors as much as possible.
- **calculations**:
  - The distance between each point in the plot and the line is known as the **residual**.
  - The loss/error function is the sum of the squares of these residuals.
  - The goal of the linear regression algorithm is to minimize this value. The sum of the squares of the residuals is known as **ordinary least squares (OLS)**.

Two-dimensional plot between the target and input feature

residual

**Resource**

[**Jolly**] Machine learning with scikit-learn Quick Start Guide, chapter 5, PacktPub, 2018



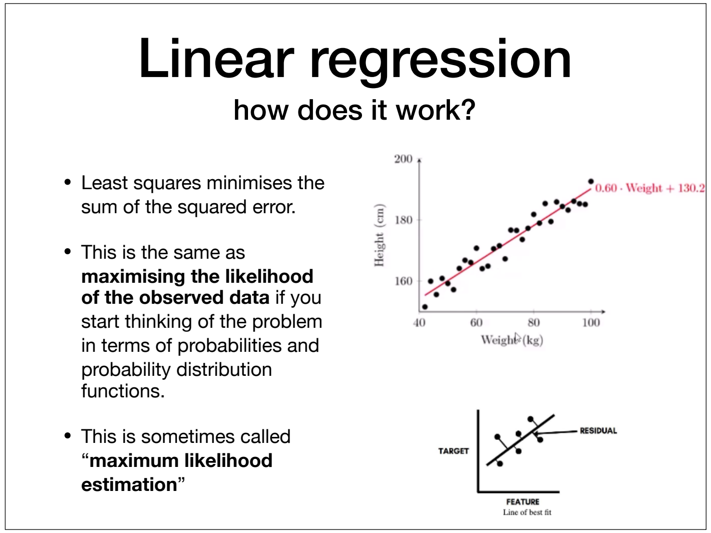
# Linear regression

## how does it work?

- Usually using “least-squares” (**OLS**)
- Minimizes the squared-error between each point and the line
- Remember the slope-intercept equation of a line? **y = mx+b**
- The **slope** is the correlation between two variables times the standard deviation in Y, all divided by the standard deviation in X
- The **intercept** is the mean of Y minus the slope times the mean of X
- **But Python wil do all that for you.**

slope **m**?
intercept **b**?

mathematics; “sum the square-distance between point and line and minimise that.” This is not complicated and will run efficiently.



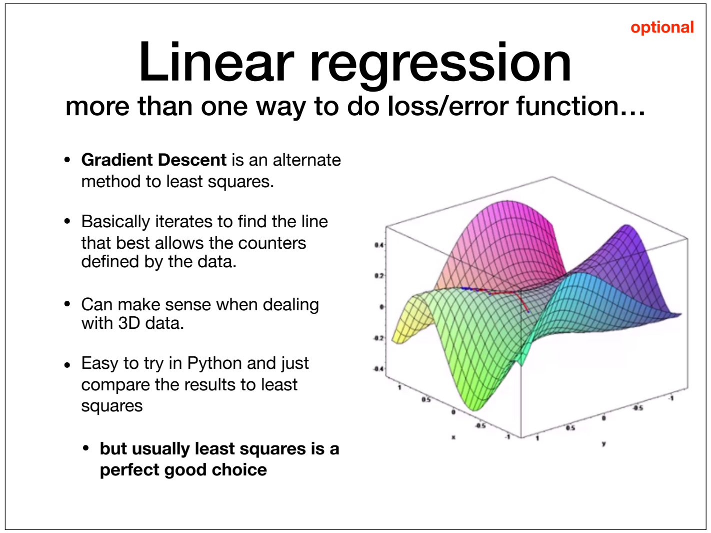
# Linear regression

## how does it work?

- Least squares minimises the sum of the squared error.
- This is the same as **maximising the likelihood of the observed data** if you start thinking of the problem in terms of probabilities and probability distribution functions.
- This is sometimes called “**maximum likelihood estimation**”

200
180
160
Height (cm)
40
60
80
100
Weight (kg)
0.60 · Weight + 130.2
TARGET
RESIDUAL
FEATURE
Line of best fit

sounding ‘smart’: talk about ‘maximum likelihood environment’, which means ‘regression’, which ‘means a point on a line which is most likely the predicted value’. Be smart too!



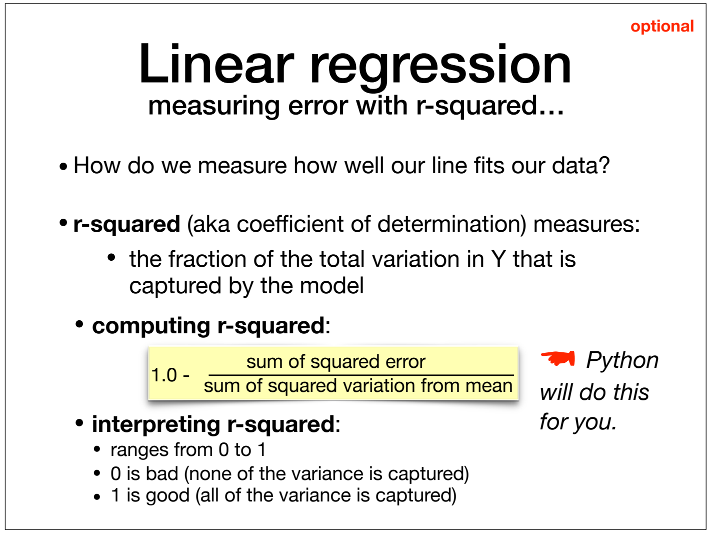
optional

# Linear regression

more than one way to do loss/error function…

- **Gradient Descent** is an alternate method to least squares.
- Basically iterates to find the line that best allows the counters defined by the data.
- Can make sense when dealing with 3D data.
- Easy to try in Python and just compare the results to least squares
  - **but usually least squares is a perfect good choice**

**Gradient Descent** being one of them, and it works best in three dimensional data, so it kind of tries to follow the contours of the data. It usually works best in higher dimensions.



optional

# Linear regression

measuring error with r-squared…

- How do we measure how well our line fits our data?
- **r-squared** (aka coefficient of determination) measures:
  - the fraction of the total variation in Y that is captured by the model
- **computing r-squared**:

$$1.0 - \frac{\text{sum of squared error}}{\text{sum of squared variation from mean}}$$

*Python will do this for you.*

- **interpreting r-squared**:
  - ranges from 0 to 1
  - 0 is bad (none of the variance is captured)
  - 1 is good (all of the variance is captured)

**How well does my line fit my data?**
Well, that's where r-squared comes in. It Is also known as the coefficient of determination again. Again, I'm trying to sound smart.

**r-squared measures the fraction of the total variation that is captured by your models.**

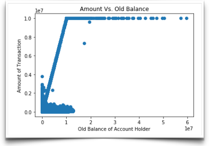
Linear Regression is a very simple technique, but there are other techniques as well, and you can use r-squared. **It's a quantitative measure of how good a given regression is to a set of data points** and then use that to choose the model that best fits your data.

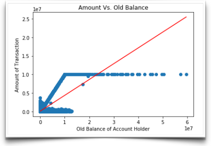
# Example

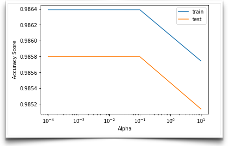
- interactive case -

**CASE: Predicting Numeric Outcomes with Linear Regression**

- predict the amount of a mobile transaction (numeric outcome) given all the other features in the dataset (*fraud_detection*).
- **Open Jupiter notebook**: casus_linear_regression_OEFENING.ipynb



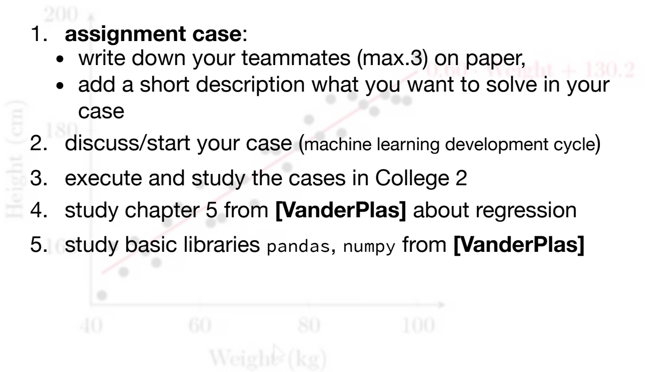




**prediction**

stel iemand weegt 120 kg, wat is zijn voorspelde lengte (height)? -> plm. 202 cm

# Exercises



1. **assignment case**:
   - write down your teammates (max.3) on paper,
   - add a short description what you want to solve in your case
2. discuss/start your case (machine learning development cycle)
3. execute and study the cases in College 2
4. study chapter 5 from **[VanderPlas]** about regression
5. study basic libraries `pandas`, `numpy` from **[VanderPlas]**

**prediction**

stel iemand weegt 120 kg, wat is zijn voorspelde lengte (height)? -> plm. 202 cm