Interdisciplinary
Center for Applied
Machine Learning

Applied
Machine Learning
Academy

Programming Languages and Frameworks for Data Science

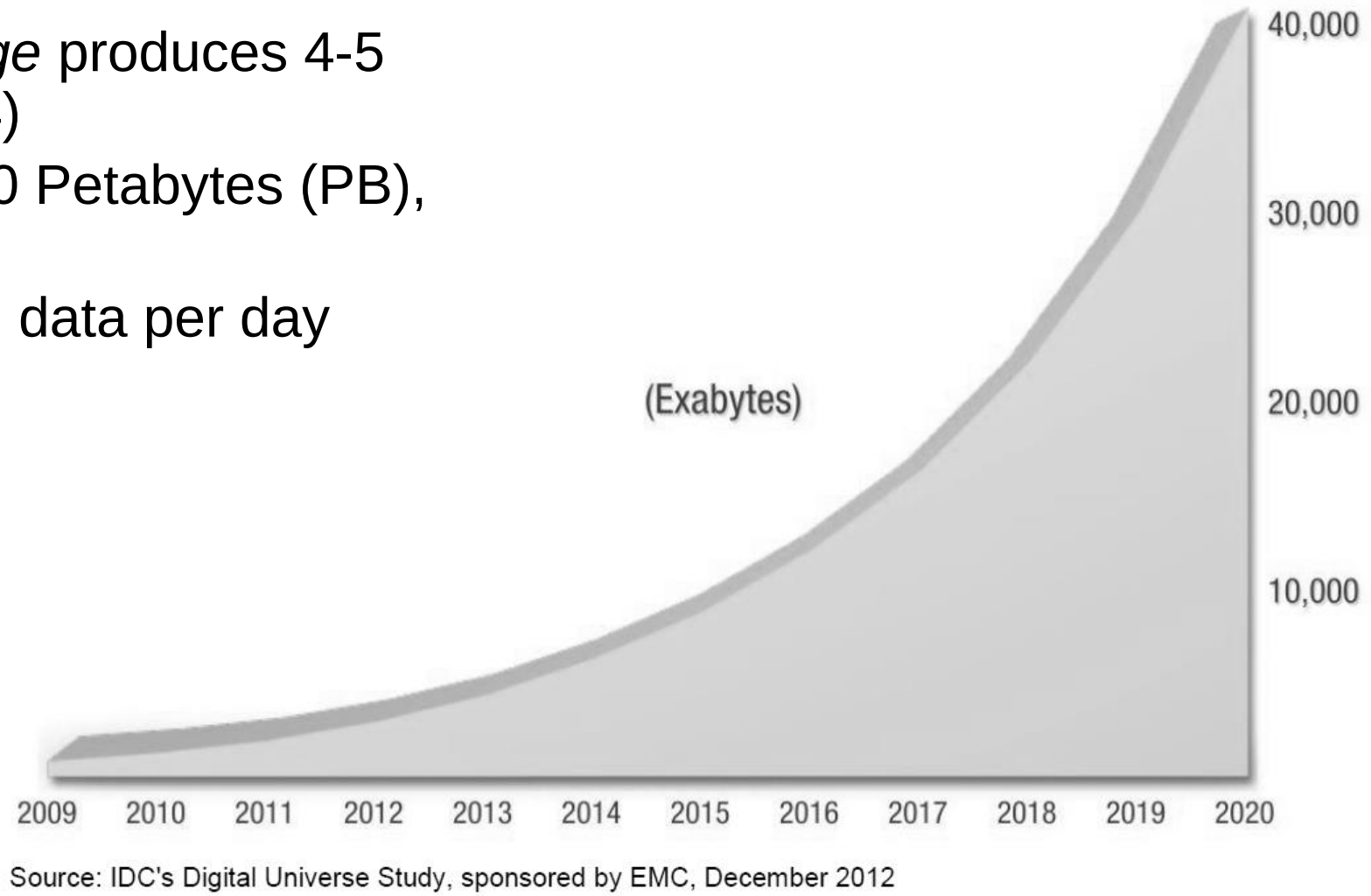
AMA / ICAML - 01.10.2019

Introduction to Data Science and Machine Learning

Why Data Science?

- Some examples
  - *New York Stock Exchange* produces 4-5 TB of data per day (2014)
  - *Internet Archive* stores 30 Petabytes (PB), i.e. 30.000 TB (2017)
  - *Google processes* 24 PB data per day (2009)

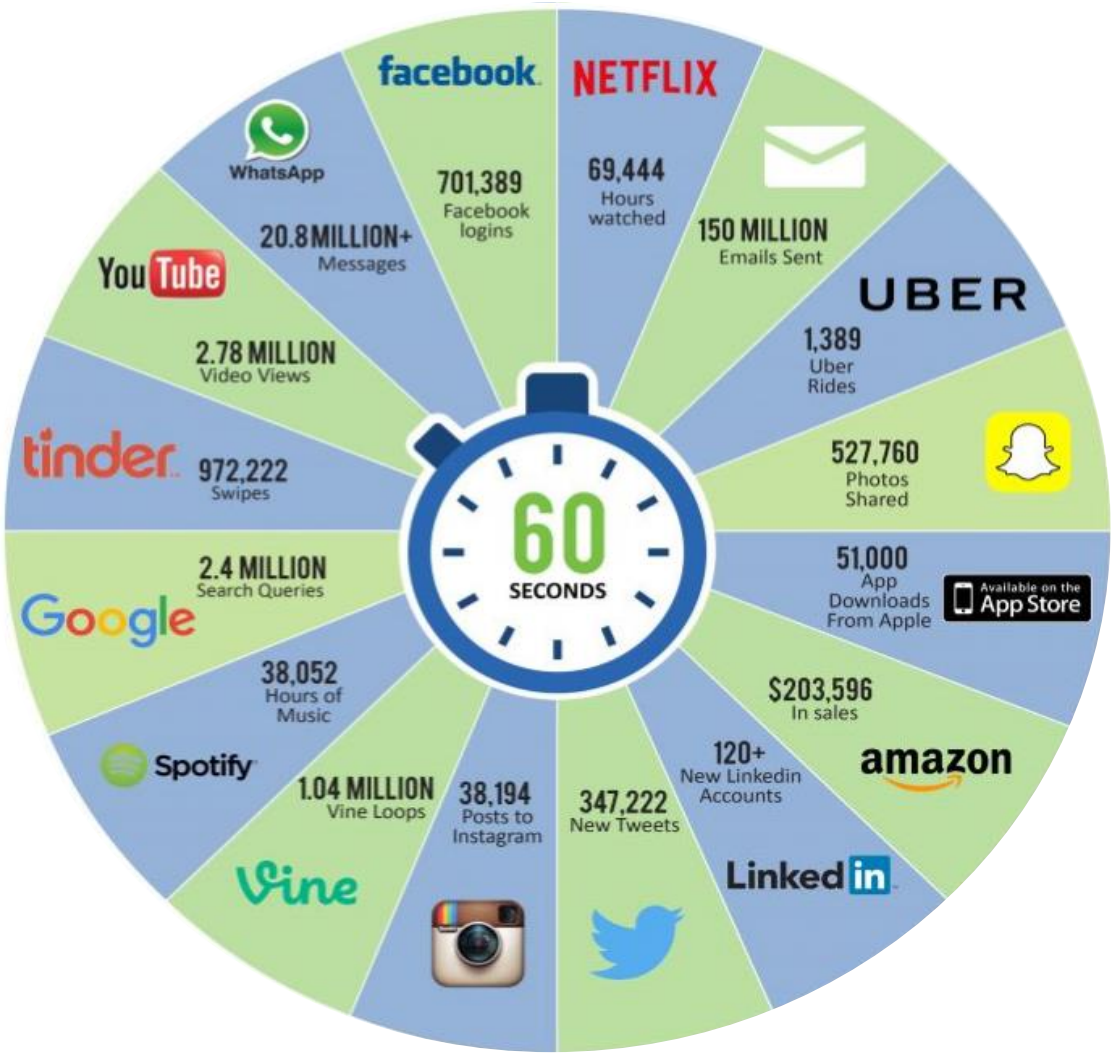
High potential, but also challenging



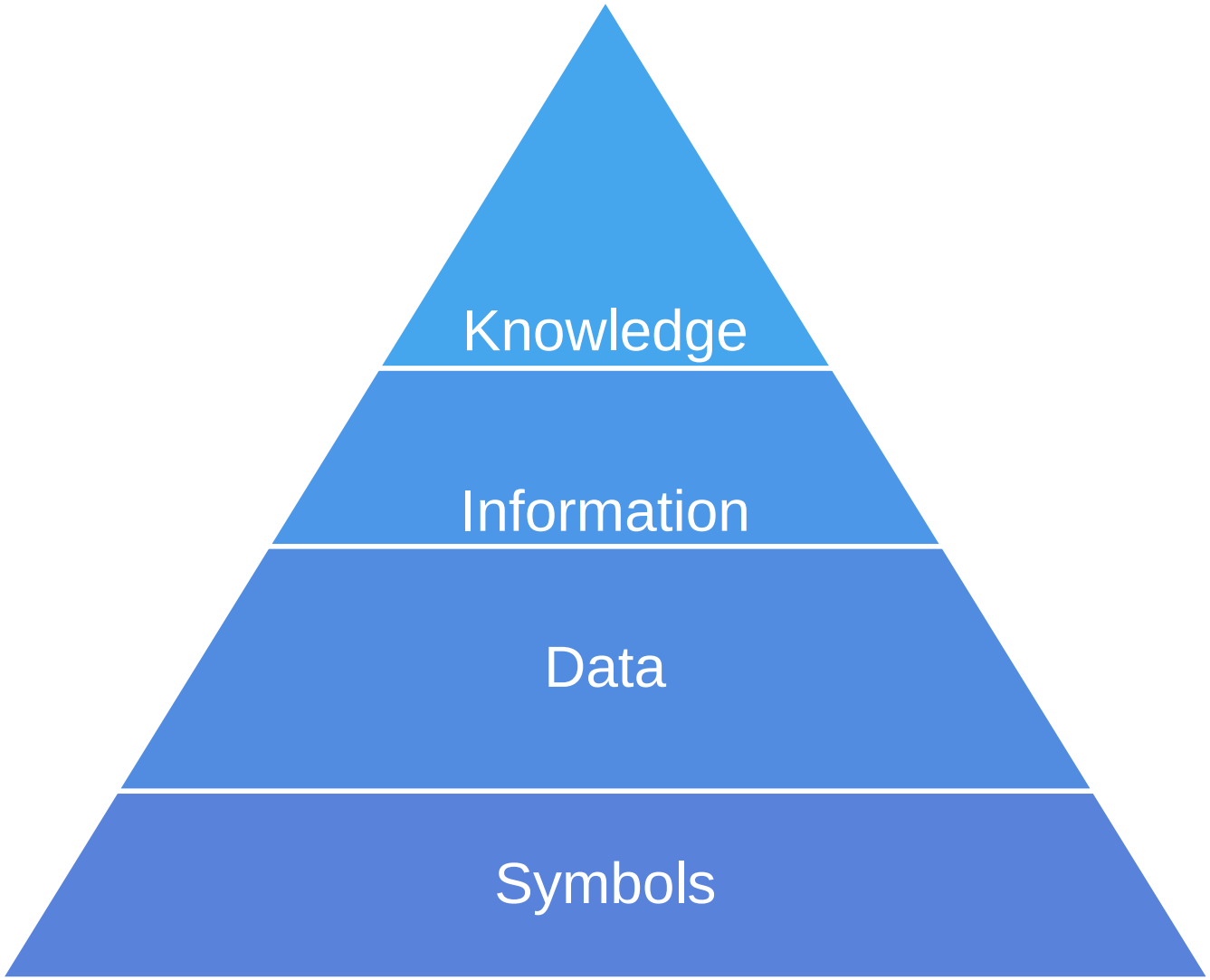
Source: IDC's Digital Universe Study, sponsored by EMC, December 2012

Exponential increase in data

- The Internet in 60 seconds



Artificial Intelligence



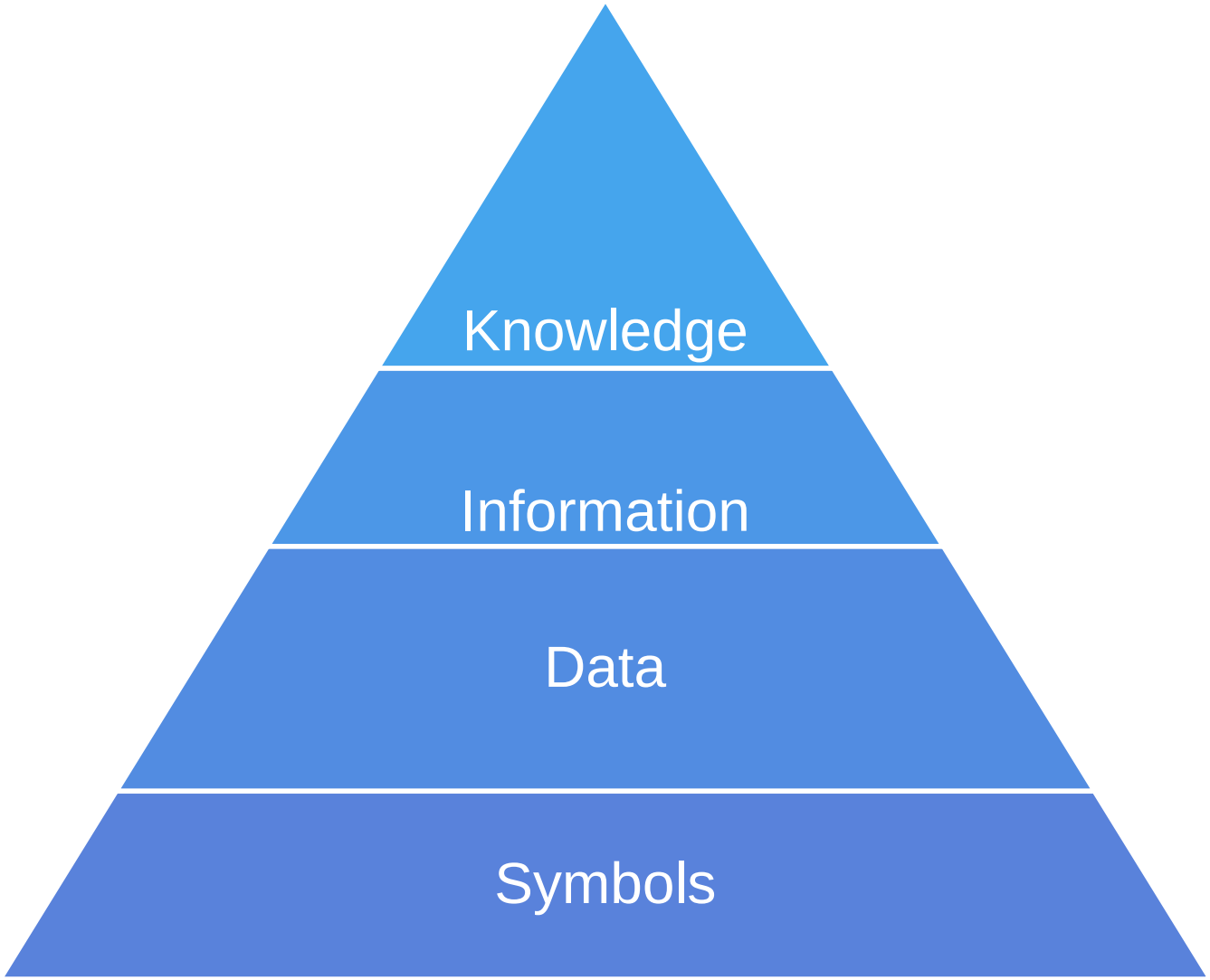
Symbolic approach

- AI = Manipulation and interpretation of symbols (also: “Knowledge”)
- Top-down: Knowledge representation, logic, inference
- Also known as “Strong AI Hypothesis” or “Physical Symbol System Hypothesis” (Newel & Simon, 1976)

Neuronal (sub-symbolic) Approach

- AI = Emulating human intelligence and its capabilities, e.g. by means of machine learning and neural networks
- Bottom-up: “Computational Intelligence”
- Also known as “Weak AI Hypothesis” (Russel & Norwig, 1995)

Artificial Intelligence



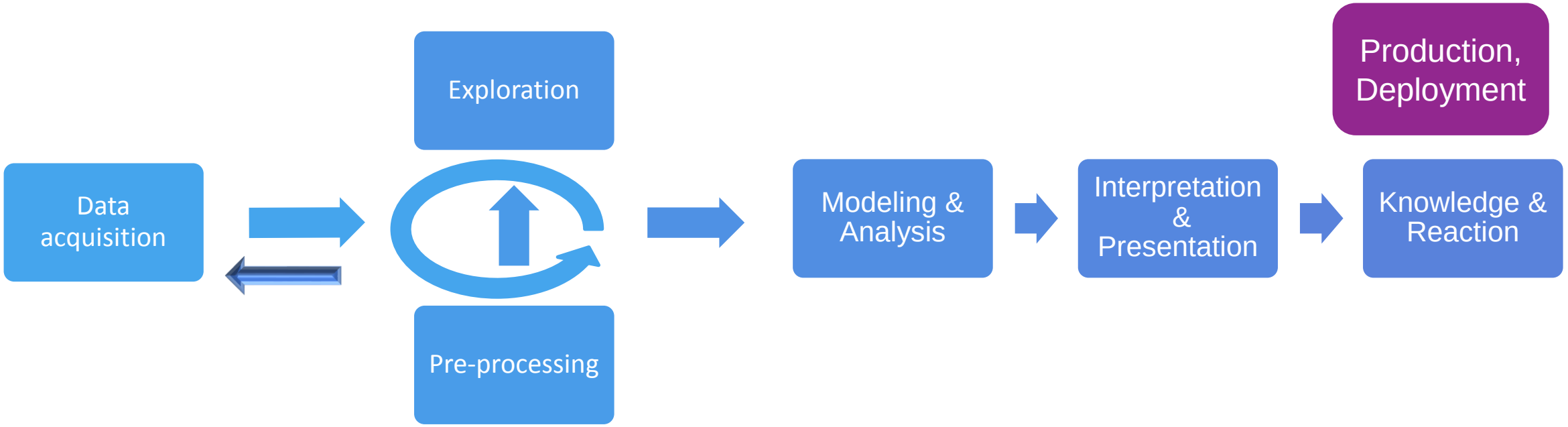
Code

- Explainability
- Test driven development
- Versioning
- Explicit model required

ML models based on data

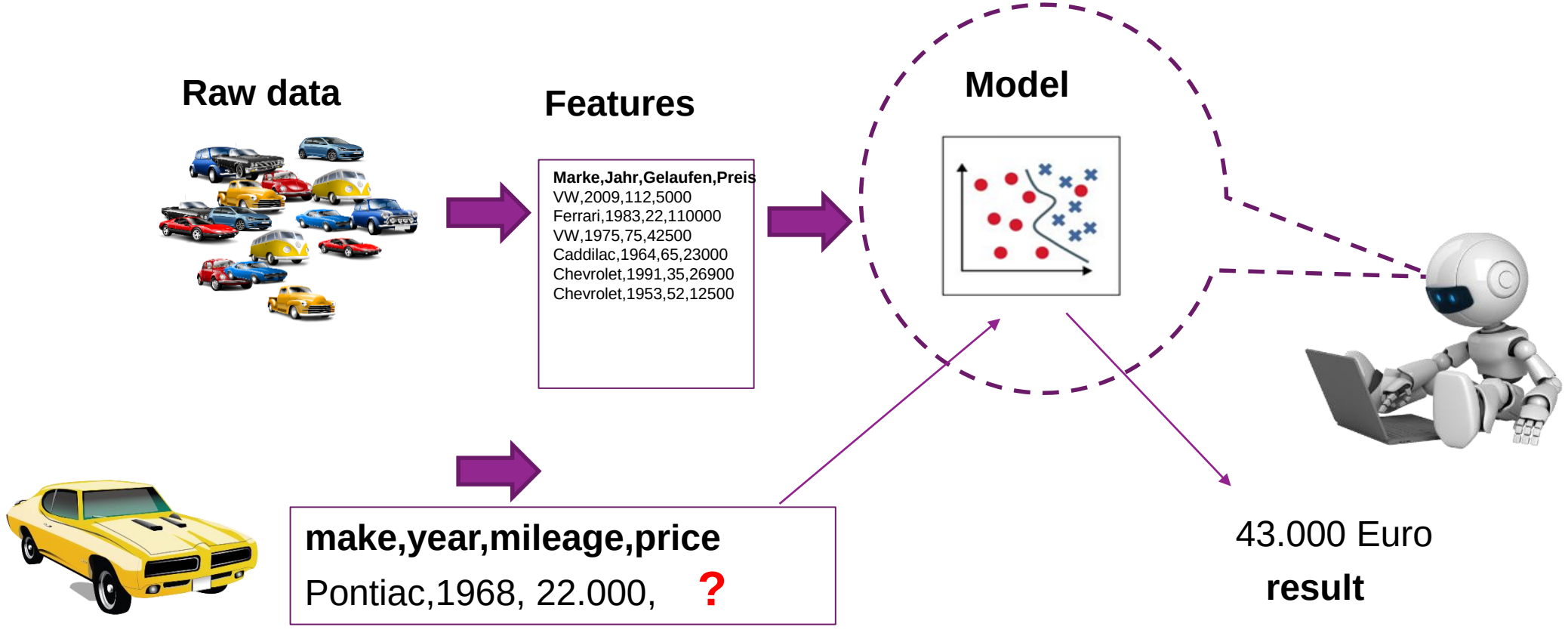
- Developer gap – labeling instead of coding
- No explicit model required
- Solution depends on data

The Data Science Process



Machine Learning: Application

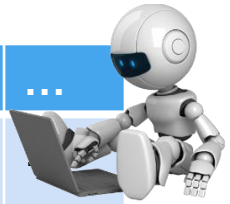
- Artificial car market expert
  - The model learns from examples. Afterwards it is able to take decisions automatically



How can a machine perceive objects? Measurements

- Every observed objects can be described with measurements (also called properties or features).
  - Measurements can be stored as, e.g., numerical or categorical values and are associated with the object

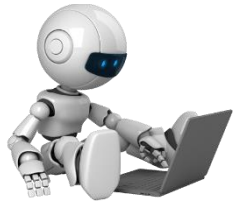
	Marke	land	Jahr	Gelaufen	Preis	…
Auto 1	VW	DE	1982	95 830	2 999,-	
Auto 2	Rover	GB	1987	88 000	3 990,-	…
Typ	Kategorie	Kategorie	Zahl	Zahl	Zahl	

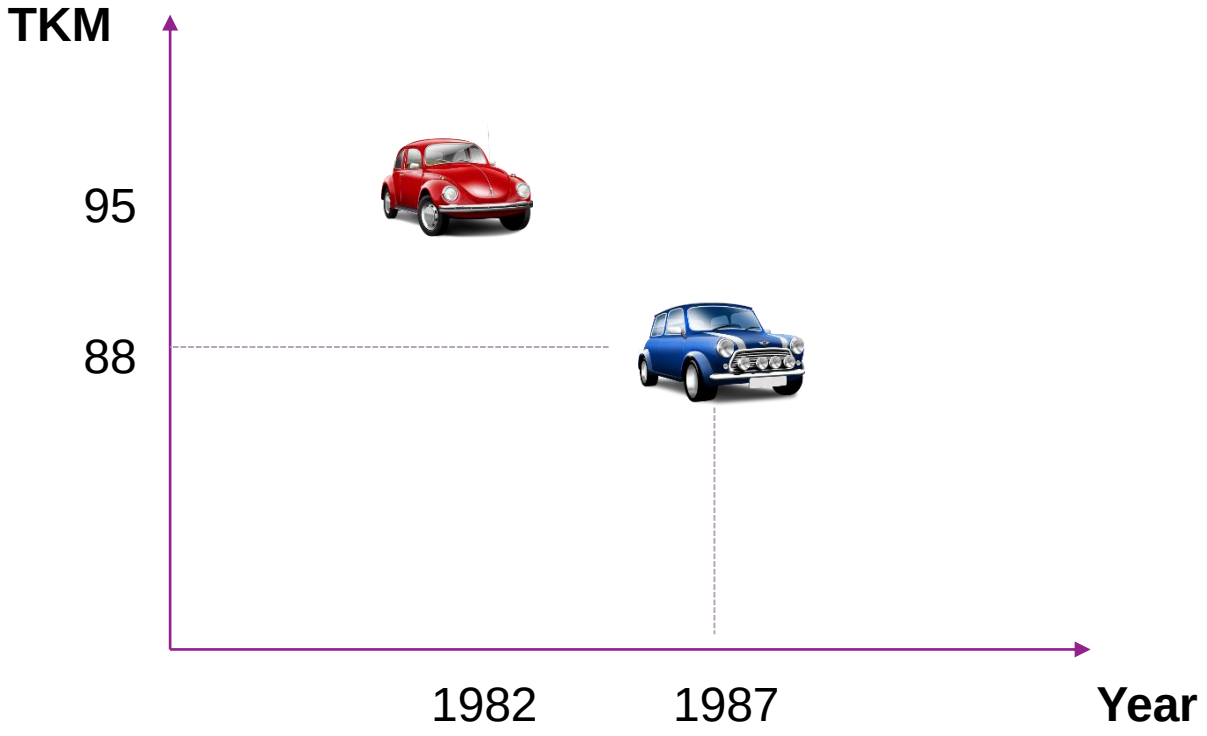


Extracting features of objects in order to..

- Properties of objects make them comparable and define a spatial position in „feature space“

➢ Describe
➢ Compare
➢ Order
➢ …

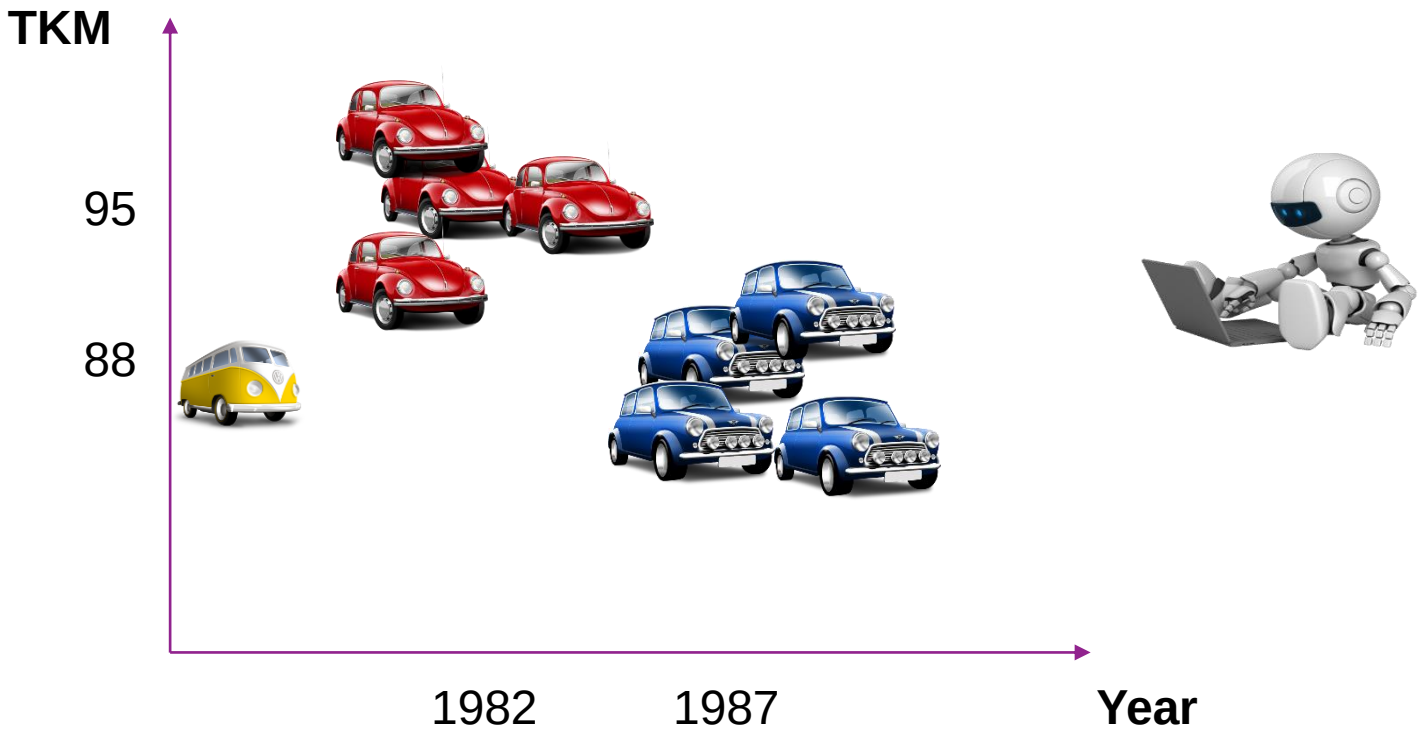




Extracting features of objects in order to..

- The more similar two objects are, the more closely they lie within feature space.

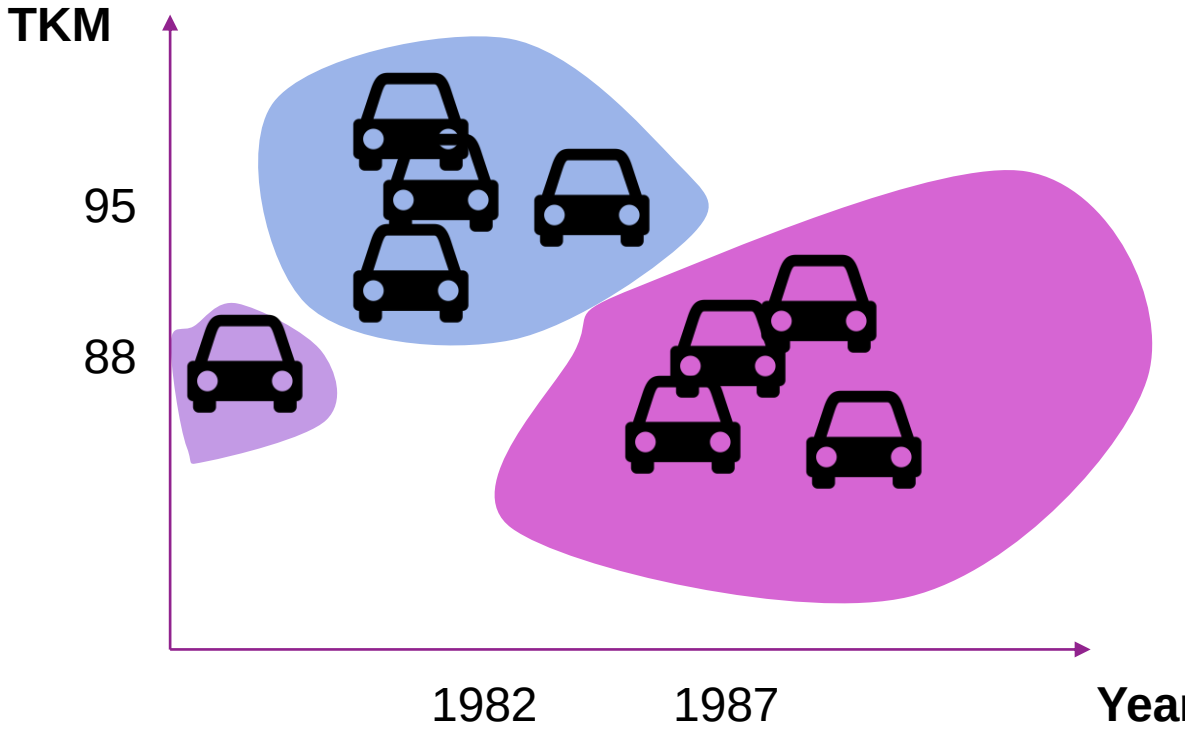
- Describe
- Compare
- Order
- Group
- Find patterns
- Find outliers

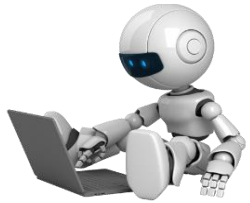


Supervised learning – clustering

- Algorithms can help to group similar objects automatically (clustering).

Finding group patterns

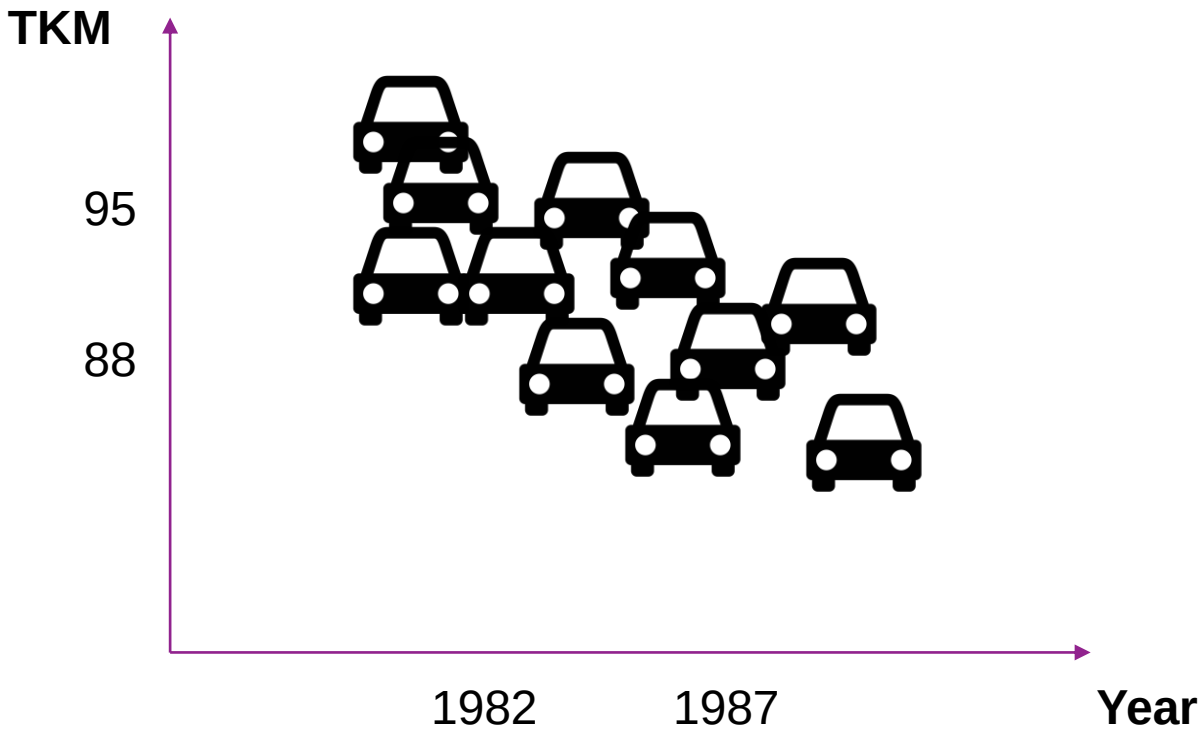


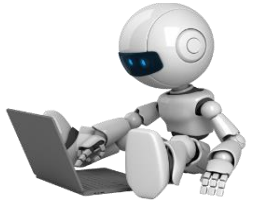


Supervised learning – classification

- Frequently, groups are hard to separate.

Find criteria to separate classes of data

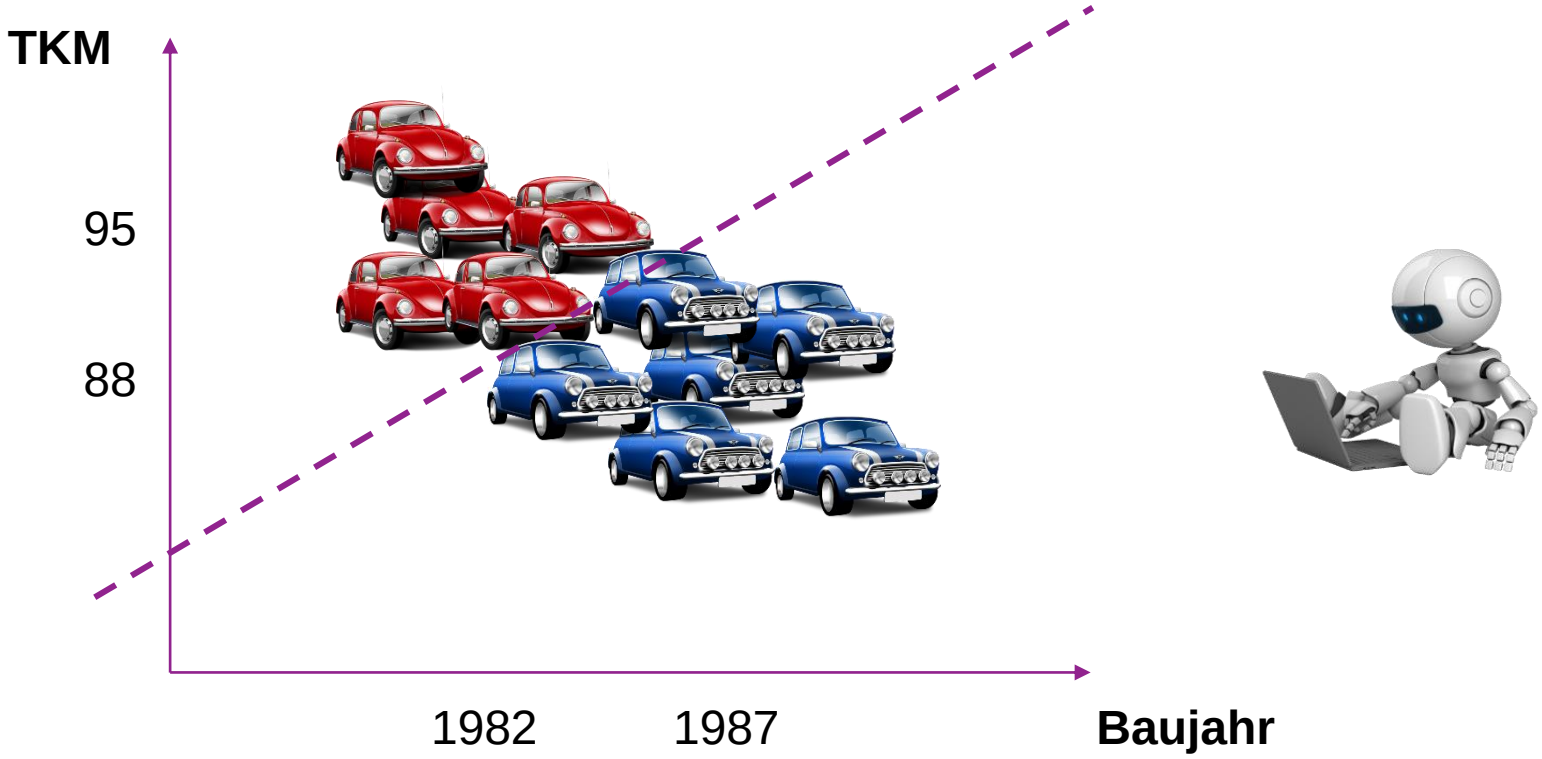




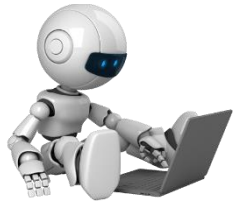
Supervised learning – classification

- Frequently, groups are hard to separate. However, if class memberships are known, the machine can compute the class boundary and assign new objects to their respective classes.

Find criteria to separate classes of data

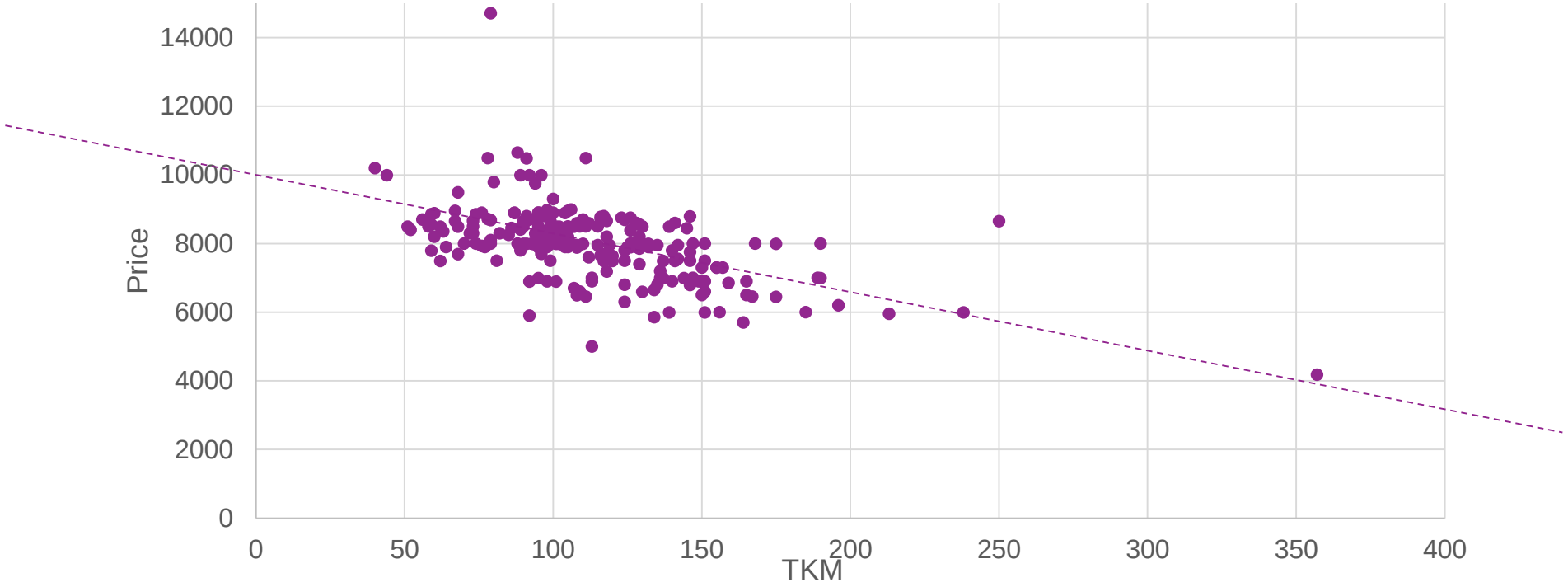


Supervised learning - regression



Golf IV TSI,2010

Feature based
value estimation

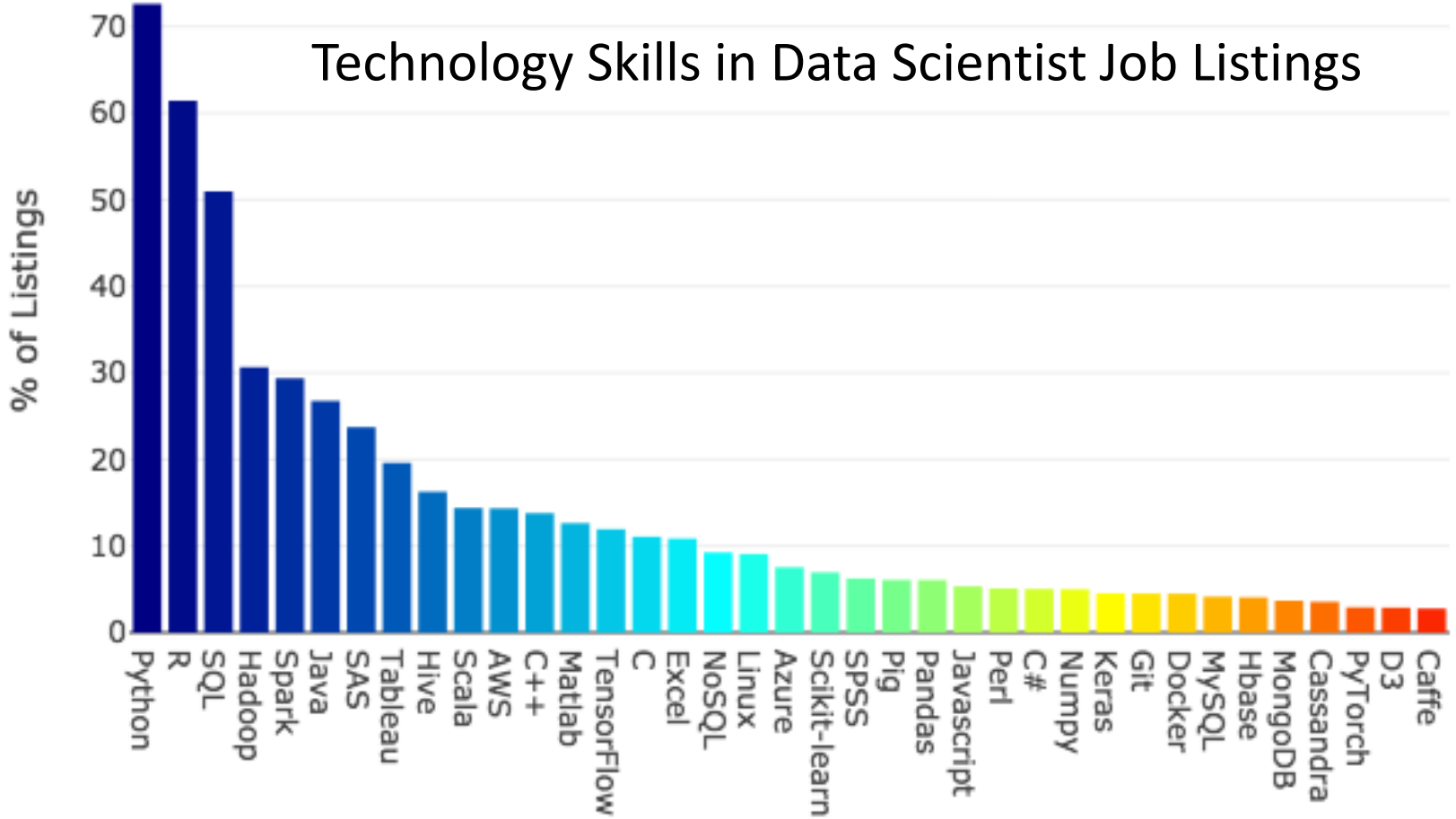


Languages for Data Science

A choice of languages for Data Science

- R
- Python
- MATLAB and Octave
- C++
- Java
- Bash (Unix)

and some more specific languages depending on the actual context.



https://towardsdatascience.com/the-most-in-demand-skills-for-data-scientists-4a4a8db896db

R

- Domain-specific language
  →allows you to focus on data science activity (not programming)

- Widely used for data science tasks in different domains
- Comes with a wide set of data sets (eases a quick start)
- Many useful packages for data science

- Focus on learning concepts of data science
- programming language just as a tool

R

Advantages

- Classical language, good documentation
- Uniform names for common actions (fit, model, predict, plot,...)
- Extremely C++ friendly (easy to extend towards high performance)
- Very good plot defaults for scientific computing
- CRAN - Peer-Reviewed source code packages for almost everything in statistical computing
- Community – Statistics domain
- Functional core

Drawbacks

- Not the easiest to start with
- Sometimes difficult to read due to complex statements

R – Data Science Process

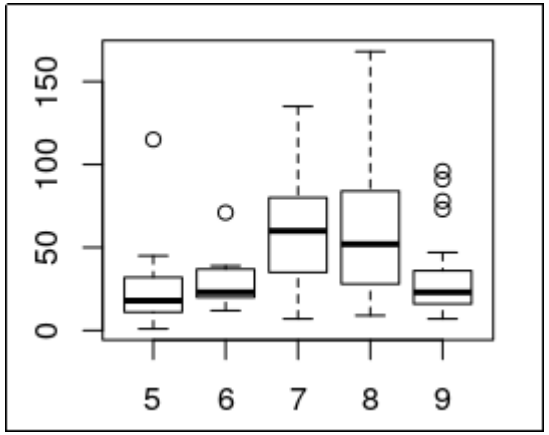
Data Acquisition	Exploration / Preprocessing	Modelling	Interpretation & Presentation	Production, Deployment
	Processing • **dplyr** • stringr • Lubridate • data.table Analysis • **base**: dataframes, stats, .. • PerformanceAnalytics	• **caret** • rminer • survival • Metrics • glmnet • randomForest	• **ggplot2** • gridExtra • xtable	• rscript • doMC

R Data Analysis

The R base package offers a comprehensive set of data analysis methods including

- Descriptive statistics
- Hypothesis testing
- Distributions
- Regression analysis
- Basic plotting functionality

```
> boxplot(Ozone ~ Month, data = airquality)
```



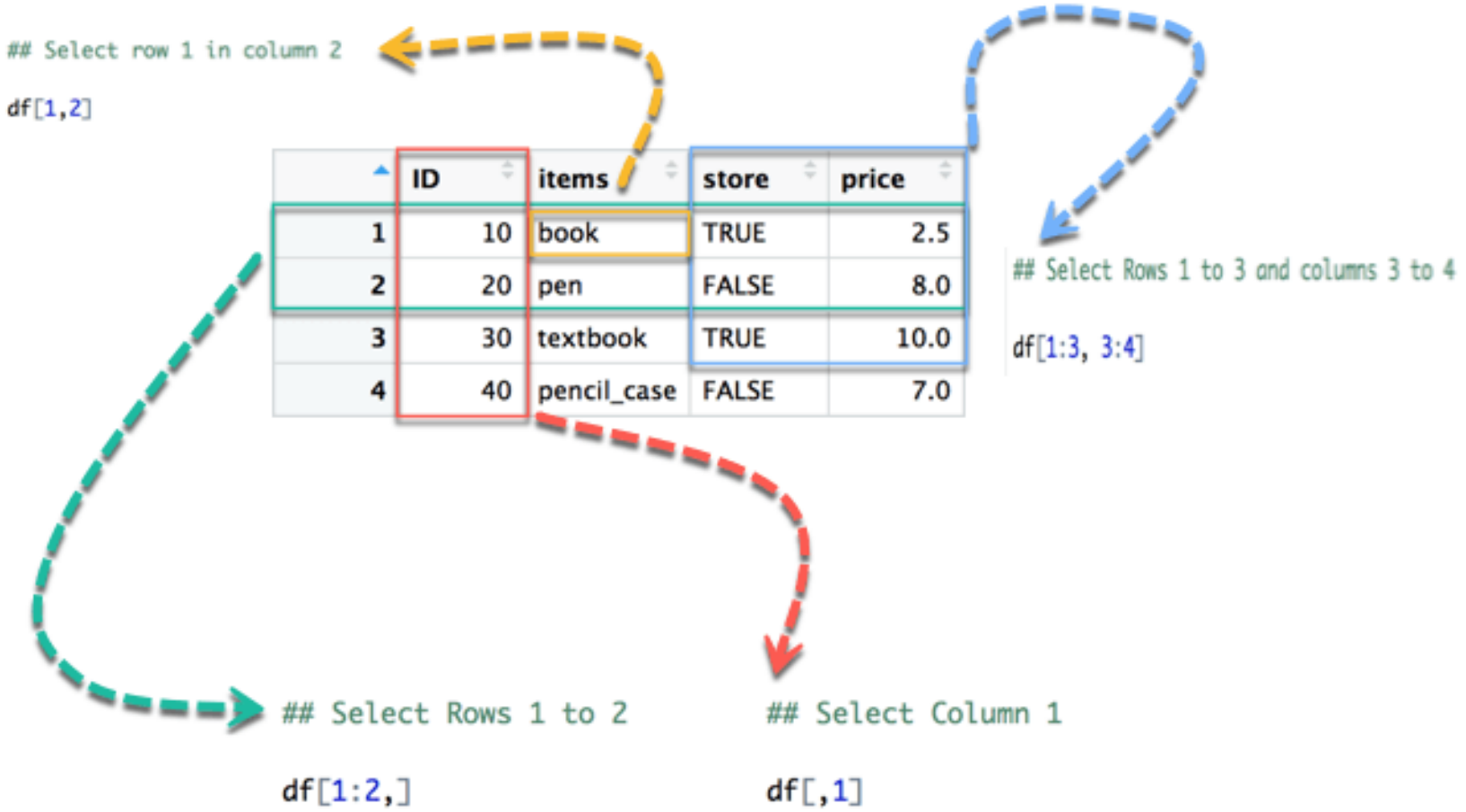

```
> wilcox.test(Ozone ~ Month, data = airquality,
+             subset = Month %in% c(5, 8))

        Wilcoxon rank sum test with continuity correction

data:  Ozone by Month
W = 127.5, p-value = 0.0001208
alternative hypothesis: true location shift is not equal to 0
```

R Data Frame

- Standard data structure in R
- Two-dimensional structure
  - Columns contain values
  - Rows contain sets of values
- Data can be numeric, factor or character type
- Offers convenient access and manipulation



Source: https://www.guru99.com/r-data-frames.html

R Processing: dplyr

Purpose: transform and summarize tabular data with rows and columns.

- Basic functions:
  - select(): filter columns
  - filter(): filter rows
  - arrange(): order rows
  - mutate(): create new columns
  - summarise(): create summary values
  - group_by(): grouping rows
- Pipe operator: “%>%”

```
msleep %>%
    group_by(order) %>%
    summarise(avg_sleep = mean(sleep_total),
              min_sleep = min(sleep_total),
              max_sleep = max(sleep_total),
              total = n())

## Source: local data frame [19 x 5]
##
##               order avg_sleep min_sleep max_sleep total
## 1     Afrosoricida 15.600000      15.6      15.6     1
## 2     Artiodactyla  4.516667       1.9       9.1     6
## 3        Carnivora 10.116667       3.5      15.8    12
## 4          Cetacea  4.500000       2.7       5.6     3
## 5       Chiroptera 19.800000      19.7      19.9     2
## 6         Cingulata 17.750000      17.4      18.1     2
## 7  Didelphimorphia 18.700000      18.0      19.4     2
```

Source: https://genomicsclass.github.io/book/pages/dplyr_tutorial.html

R Modelling: caret

- The caret package streamlines creating predictive models.
- The package offers tools for:
  - data splitting
  - preprocessing
  - feature selection
  - model tuning using resampling
  - variable importance estimation
- A large number of models contained in the package

```
fitControl <- trainControl(## 10-fold Cross-Validation
                           method = "repeatedcv",
                           number = 10,
                           ## repeated ten times
                           repeats = 10)

gbmFit <- train(Class ~ .,
                data = training,
                method = "gbm",
                trControl = fitControl,
                ## This last option is actually one
                ## for gbm() that passes through
                verbose = FALSE)

gbmFit
```

R Visualization: ggplot2

- One of the most widely used visualization packages
- Enables creating sophisticated visualisations using grammar of graphics
  - Graphs are broken up into sematic components

```
md <- median(abs(data$cal))

ggplot(dat, aes(abs(data$cal)))  +
    geom_histogram(alpha=alpha, fill=fill,color=bordercolor,size=bordersize, aes(y=..count..))  +
    xlab("Calories") +
    ylab("Frequency") +
    annotate("text", x = 400, y = 1400, label = paste("Median:", round(md,digits = 2))) +
    geom_vline(aes(xintercept=cdat11),linetype="dashed",color="black", size=1)
```