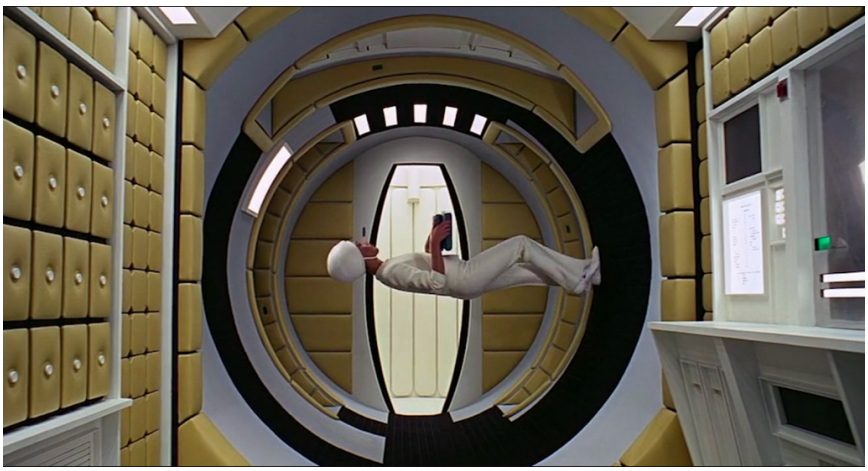
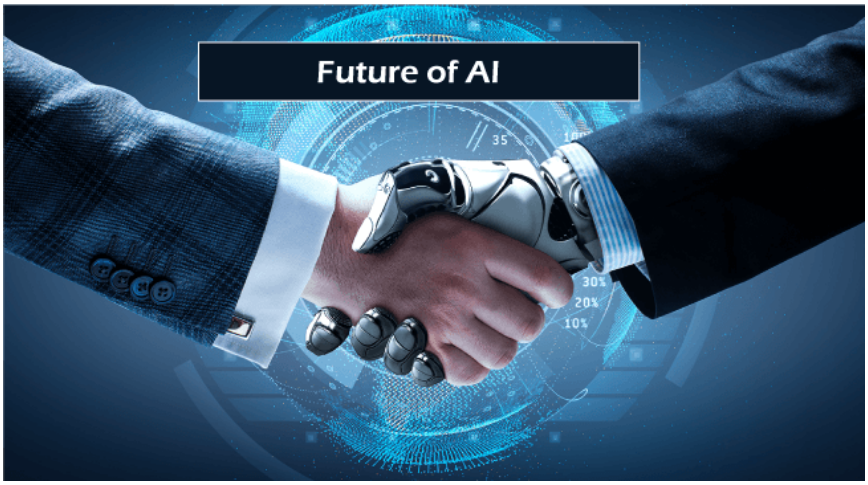
INTRODUCTION TO MACHINE LEARNING

Marcus Birkenkrahe

January 10, 2023



README



The purpose of today's machine learning (ML) is to assist us with making sense of the world's massive data collections.

1. Origins, applications and pitfalls of ML
2. How computer use models to transform data into knowledge
3. Steps to math an ML algorithm to your data
4. Simple R examples

Origins of ML

Figure 1: Image: Assassin's Creed Origins (2017)

1. **Statistical modeling** (17th century)[1]
2. **Machine learning** (1950s)[2]

[1] Statistical methods make assumptions about the nature of the data. If these assumptions are violated the models built with the data are inaccurate. Systematically developed in the 17th and 18th century, popular since the rise of computers in the mid-20th century.

[2] For ML, assumptions about data distributions and variable independence are not a concern. ML is part of the field of Artificial Intelligence (AI) - computers who make their own decisions. Popular since 1959 (Arthur Samuel: ML is the ability to learn without being explicitly programmed.). Subfields include deep learning = ML with artificial neural networks; reinforcement learning = reward-based machine learning

3. **Knowledge discovery** (1980s)[3]
4. **Data mining** (1990s)[4]
5. **Data science** (2010s)[5]

What about "intelligence"?

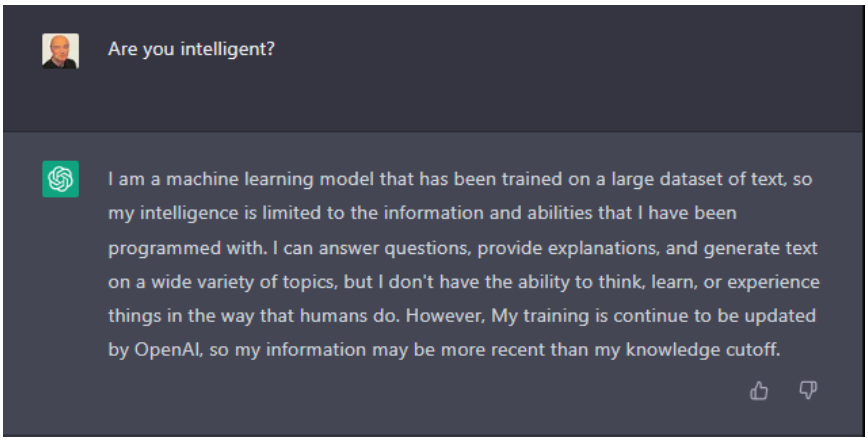


Figure 2: ChatGPT's answer to "Are you intelligent?"

[3]Knowledge discovery in databases (KDD) was coined in 1989 to emphasise that knowledge can be derived from data-driven discovery. Frequently used interchangeably with data mining. Includes not just the pattern search but also methods for extracting, preparing and using data.

[4]Data mining is the process of using one or more machine learning algorithms to find patterns or structure in data. The patterns may be: a set of rules, a graph or network, a tree, one or more equations, etc. The applications can be part of a visual dashboard or as simple as a list. Popular since ca. 1995.

[5]Data science refers to the process of extracting meaningful knowledge from data, using methods from statistics, computer science, database management and more. ML is not required for data science but it is often used - e.g. when fitting a trendline to a dataset. Popular since ca. 2012. Subfields include data engineering, process mining, and machine learning.

Uses and abuses of ML

- Deep Blue defated Kasparov (1997), Watson won Jeopardy (2011)
- Machines are pure horsepower without direction - they "need a human to motivate the analysis and turn the result into meaningful action" - like a hound with a human.
- Interesting analysis of Kasparov vs. Deep Blue (Anderson, 2017)
  - A coding bug may have misled Kasparov to overestimate Deep Blue
  - Conspiracy? Deep Blue may have have been a "Mechanical Turk"
  - Illustrates the difference between man and machine

ML successes

Many different uses, many different models:

- **Identification** of unwanted spam messages in email
- **Segmentation** of customer behavior for targeted advertising
- **Forecasts** of weather behavior and long-term climate changes
- **Prevention** of fraudulent credit card transactions
- **Estimation** of actuarial financial damage of natural disasters

Figure 3: Inside of a Tesla car driving autonomously

- **Prediction** of popular election outcomes
- **Autonomous** vehicles: auto-piloting drones and self-driving cars
- **Optimization** of energy use in homes and office buildings
- **Projection** of areas where criminal activity is most likely
- **Discovery** of genetic sequences linked to diseases

Limits of ML

- Little flexibility outside of strict parameters and no common sense
- Consequences of releasing an algorithm hard to predict[6]
- Inability to make simple inferences about logical next steps (e.g. repeatedly served banners on ecommerce sites)
- Random epic failures: handwriting recognition, 1994

[6] See Loizos (Dec 9, 2022): "Is ChatGPT a 'virus that has been released into the wild'?" - 2019 interview with Sam Altman (OpenAI)

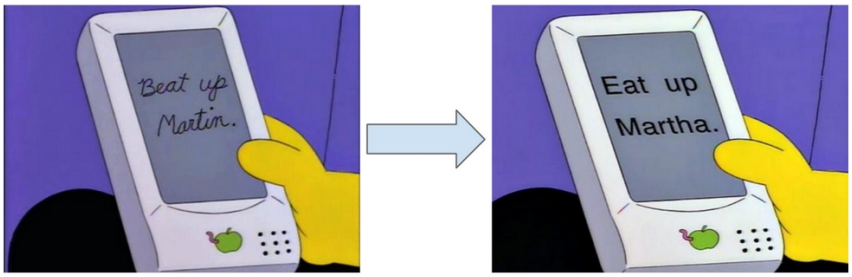


Figure 4: Lisa on Ice, The Simpsons, 20th Century Fox (1994)

- Auto-correct failures (ML insists on what you once wanted/were)
- Natural language processing is still very difficult (but: ChatGPT)
- Alas, we often adapt to the limited abilities of our machines

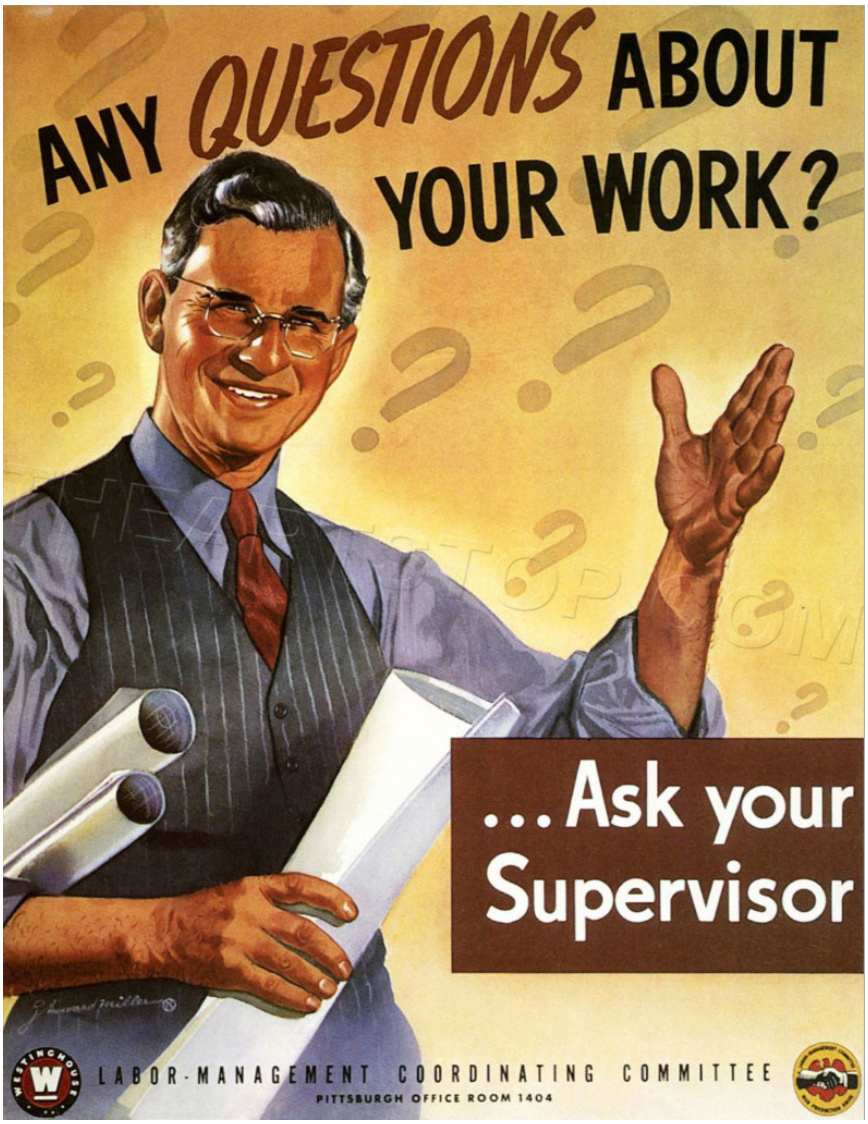
ML ethics

Figure 5: Lisa on Ice, The Simpsons, 20th Century Fox (1994)

- Like any tool, it can be used for "good" or for "evil"

- Associated **legal** issues and social norms are still uncertain
- Issues include **privacy** rights of customers
- Handing **critical** operations (e.g. airport control) to machines
- Relying on ML in **life-or-death** situations (medical diagnosis)
- **Blindly** applying ML analysis results to make decisions
- **Perpetuating** discrimination based on race or gender
- **Reinforcing** negative stereotypes
- **Anonymizing** data is difficult because ML is good at finding you
- **Regulation**, e.g. EU's General Data Protection Regulation (GDPR)
- ML can be used for fake news, or **misguiding** autonomous systems

Extended example: supervised learning



Process:

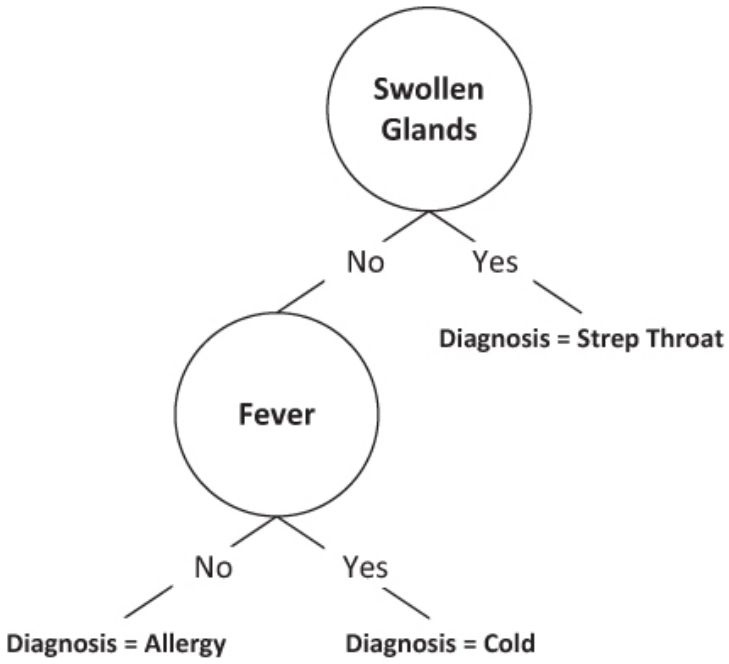
1. Build a classification model from known data instances
2. Test model to classify newly presented unknown data instances
3. Translate model into algorithmic production rules

Building a model from training data

- Dataset: hypothetical training data for a disease diagnosis

Patient ID#	Sore Throat	Fever	Swollen Glands	Congestion	Headache	Diagnosis
1	Yes	Yes	Yes	Yes	Yes	Strep throat
2	No	No	No	Yes	Yes	Allergy
3	Yes	Yes	No	Yes	No	Cold
4	Yes	No	Yes	No	No	Strep throat
5	No	Yes	No	Yes	No	Cold
6	No	No	No	Yes	No	Allergy
7	No	No	Yes	No	No	Strep throat
8	Yes	No	No	Yes	Yes	Allergy
9	No	Yes	No	Yes	Yes	Cold
10	Yes	Yes	No	Yes	Yes	Cold

- Patient 1 has a sore throat, fever, swollen glands, is congested and has a headache. He was diagnosed with strep throat.
- A *decision tree* can be used to generalize a set of input instances as shown and transform it into rules.
- To generalize, we must make assumptions about the relative importance of attributes and their relationship
- For example:
  - If a patient has swollen glands, the diagnosis is strep throat
  - If a patient does not have swollen glands and a fever, it's a cold
  - If a patient does not have swollen glands nor a fever, it's allergy



- The attributes *sore throat*, *congestion* and *headache* do not enter our diagnostic prediction

Testing the model on unknown instances

- Moving on to a new data set with unknown classification, i.e. no diagnosis
- Use the decision tree to classify the first two instances:

Patient ID	Sore Throat	Fever	Swollen Glands	Congestion	Headache	Diagnosis
11	No	No	Yes	Yes	Yes	?
12	Yes	Yes	No	No	Yes	?
13	No	No	No	No	Yes	?

- Patient 11 has swollen glands but no fever => strep throat
- Patient 12 has no swollen glands but fever => cold

Translate model into production rules

- General form of a *production rule* looks like pseudocode[7]:

```
IF antecedent condition
THEN consequent conditions
```

- The three *production rules* for the decision tree:

```
IF swollen glands = YES
THEN diagnosis = strep throat

IF swollen glands = No & Fever = Yes
THEN diagnosis = cold

IF swollen glands = No & Fever = No
THEN diagnosis = allergy
```

- Testing the rules on patient 13 yields: diagnosis = allergy

How machines learn

- Unlike humans, machines need explicit conditions and instructions literally down to the letter (ML does not change that completely)

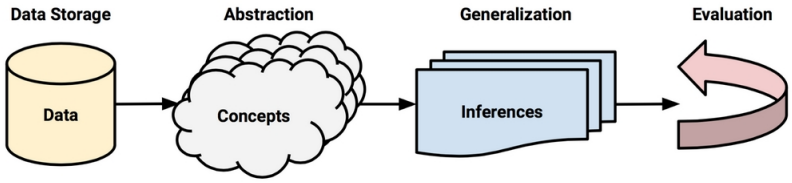
☐ What's the effect for humans when making everything very explicit? **Does explicitness help or hinder human learning?**[8]

- To be a strong data scientist / ML practitioner requires solid understanding of **how the learning algorithms work**

- Basic ML process:

[7] This is exactly what pseudocode is: natural language without the constraints of syntactical rules. What used to be helpful in the past could in the future well become the standard for programming, cp.

[8] How did you "learn" to close a door? Did your mom give you a long lecture about the physics, the difference in temperature on either side of the door, and about the different ways to grip and turn the handle? What happens when you encounter a different handle you have not seen yet? What if you encounter a door that has no handle?



☐ How does the extended diagnosis example fit in this scheme?[9]

Data storage = observe + memorize + recall

- **Data storage** utilizes observation, memory, and recall to provide a factual basis for further reasoning

☐ Storage needs to take software and hardware conditions into account - how?[10]

- You need to store raw data selectively - **more data** does not necessarily mean **more information** (too much data can obscure what you're looking for) and carries a performance overhead

[9] Diagnosis ML example: (1) Data storage: raw patient data; (2) Abstraction: table of attributes and records; (3) Generalization: rules from known diagnosis; (4) Evaluation: prediction of unknown diagnosis

[10] Storage needs to take software and hardware conditions into account: (1) performance: access speed; (2) data organisation, e.g. relational or non-relational data; (3) missing or otherwise contaminated data; (4) R: all data is in memory (space issue); (5) SQLite: one ffile non-concurrent access (security/usability issues)

- Remember studying for an exam - do you gorge yourself on all available details or do you select questions and answers that were discussed in class?[11]

Nile example: data storage

- To run the code below, open tinyurl.com/2mnv425w, save the file to `1_ml_practice.org` and open it in Emacs.
- Once you're done, upload the completed file to Canvas:
- Example: the following numbers come from R's `Nile` data set:

```
1120 1160 963 1210 1160 1160 813 1230 1370 1140
995 935 1110 994 1020 960 1180 799 958 1140
1100 1210 1150 1250 1260 1220 1030 1100 774 840
874 694 940 833 701 916 692 1020 1050 969
831 726 456 824 702 1120 1100 832 764 821
768 845 864 862 698 845 744 796 1040 759
781 865 845 944 984 897 822 1010 771 676
649 846 812 742 801 1040 860 874 848 890
744 749 838 1050 918 986 797 923 975 815
1020 906 901 1170 912 746 919 718 714 740
```

- To extract the data from the data set (already stored in R):

```
write(x=Nile,
      file="../data/Nile.txt", # Unix-style forward slash
      ncolumns=1,
      sep=" ")
```

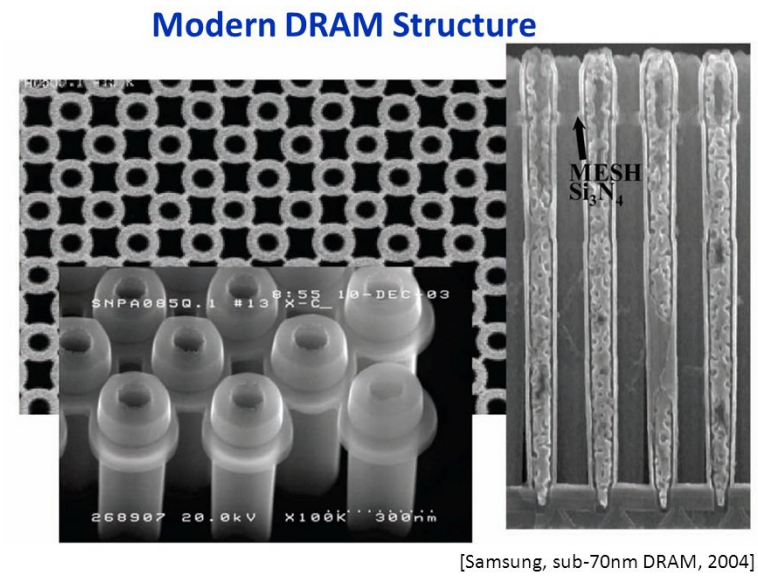
- The values are stored as a text file `Nile.txt` of size 440 byte, which means 440 * 8 = 3520 bits, or binary value capacitors:

```
shell(cmd="DIR ..\\data\\Nile.txt") # escaped Windows backward slash
```

[11] In fact, human learning is poorly understood: if you have an eidetic memory (Sheldon-Cooper-style), storing everything may be a valid strategy. I don't think I have that but I still like to fill myself up with seemingly "irrelevant" data - and I trust my guardian angel, or my intuition, or whatever you will, to pull the proverbial rabbit out of a hat when needed. This has often worked for me!

```
 Volume in drive C is OS
 Volume Serial Number is 0654-135C

 Directory of c:\Users\birkenkrahe\Documents\GitHub\ds2\data

01/10/2023  10:53 AM               530 Nile.txt
               1 File(s)            530 bytes
               0 Dir(s)  311,921,278,976 bytes free
```

- When on disk, `Nile.txt` is stored in non-volatile memory (it's permanent). When it is loaded into R (or another shell program), it is represented as RAM (Random Access Memory), physically realized as a capacitor that is charged (1) or uncharged (0) (source).



- You can look at the text file using `notepad`:

```
shell(cmd="notepad ..\\data\\Nile.txt")
```

Abstraction = transform + train

Image: Magritte, La Trahison Des Images[12]

- **Abstraction** involves translating stored data into broader representations and concepts

☐ Abstraction needs to take available computing data structures into account - how?

- The nature of a "representation" is that it is **not the original** - for ML, recognition is more important than reality: the AI is not trying to build a world, but to translate it into something it can "see"

Nile example: transformation

- `Nile` example: earlier, we stored integer numbers in memory. A convenient representation in R involves choosing a **data structure** and transforming the numbers into it

[12] René Magritte's painting "The Treachery of Images" illustrates the idea of a representation: *"Ceci n'est pas une pipe"* because it's an image of a pipe, and not the pipe itself.

- We read the text data from file using the R function `read.table` and store them in a time series using the R function `ts`:

  1. read the text file `read.table` as a `data.frame`
  2. remove column name with `colnames`
  3. create time series with `ts` from data frame

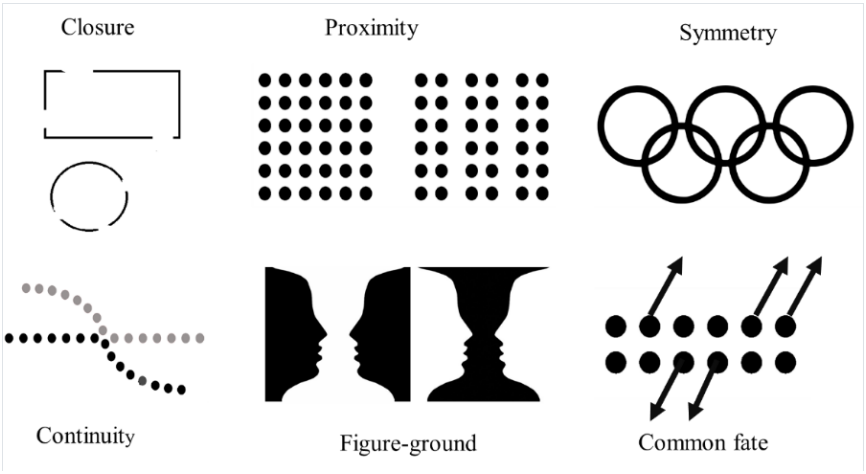
```
 nile_df <- read.table(
  file="../data/Nile.txt",   # read from text file
  sep=" ",                  # entries separated by empty space
  header=FALSE)             # no 1st row with attribute information
colnames(nile_df) <- NULL
nile_ts <- ts(nile_df,start=1871)
```

- The transformed data set contains additional information that was not present in the numbers themselves. We have used additional information (about the origin of the data) and R's time series data structure.

```
str(nile_ts)
class(nile_ts)
```

```
 Time-Series [1:100, 1] from 1871 to 1970: 1120 1160 963 1210 1160 1160 813 1230 1
 - attr(*, "dimnames")=List of 2
  ..$ : NULL
  ..$ : NULL
[1] "ts"
```

Modeling



Source: Hosseini, Hytönen, Kinnunen (2022)[13]

- When a machine creates a **Knowledge representation**, it summarizes stored raw data using a **model**, an explicit description of the patterns within the data
- A model represents an idea greater than the sum of its parts (also: "The whole is greater than the sum of its parts")
- Machines, unlike humans, cannot comprehend these Gestalt patterns as a whole, they can only sequentially process the components of a pattern[14].
- There are many different types of models, including:
  - Mathematical equations
  - Relational diagrams, such as trees and graphs
  - Logical if/else rules (conditional structures)
  - Groupings of data (clusters)

[13] In visual perception, the idea of a model summarizing data is illustrated by the six Gestalt (German for "shape") principles: each of them implies not just the pixels of the image but a pattern that leads to a human process of perceiving more than just the pattern itself.

[14] This was one of the critiques of AI by philosopher Hubert Dreyfus (see Wikipedia here and here for a graph representation).

- Typically , the machine does not pick the model - it is picked by a human depending on the learning task and the type of data available

Nile example - modeling

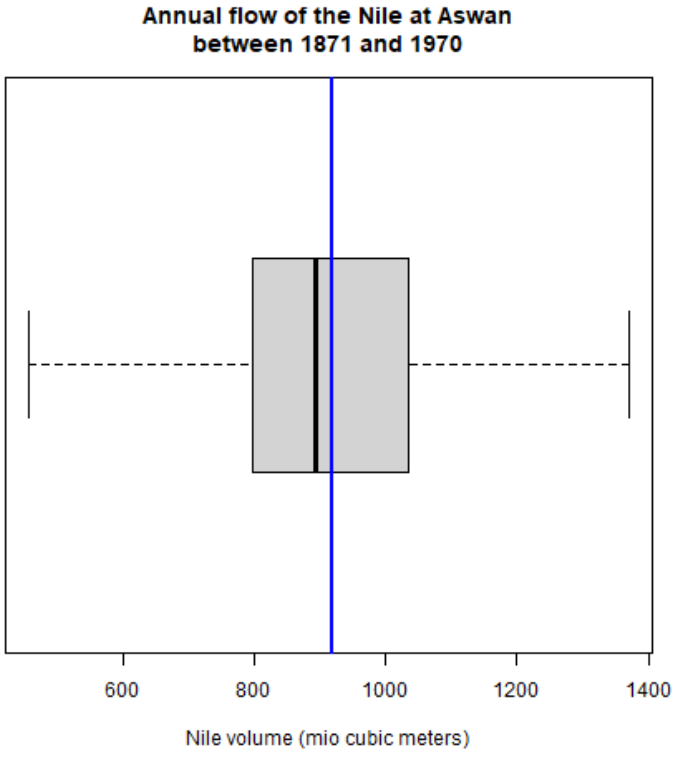
- As an example of statistical inference, we use the time series data of `Nile` to create a statistical model
- In R this is easily achieved with the `summary` function

```
data(Nile)  # add the built-in Nile dataset to the session
ls()  # show all R objects in the current session
summary(Nile) # 5-point summary + sample average
```

```
[1] "Nile"    "nile_df" "nile_ts"
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  456.0   798.5   893.5   919.4  1032.5  1370.0
```

- To visualize this model, you can use `boxplot` (and `abline` to add the `mean`):

```
boxplot(Nile,
las=1, # reorient x-axis labels
horizontal=TRUE, # show boxplot horizontally
main="Annual flow of the Nile at Aswan\nbetween 1871 and 1970",
xlab="Nile volume (mio cubic meters)")
abline(v=mean(Nile),  # draw a vertical line
       col="blue",    # paint line blue
       lwd=2)         # double line width
```



- The *generic* function `summary` collapses the abstraction (time series representation) into a statistical summary
- That `summary` is *generic* is relevant because it means that it can deal with many different abstractions (and models, too):

```
methods(summary)
```

```
 [1] summary,ANY-method                summary,DBIObject-method
 [3] summary.Anova.mlm*                summary.aov
 [5] summary.aovlist*                  summary.aspell*
 [7] summary.bcnPowerTransform*        summary.bcnPowerTransformlmer*
 [9] summary.boot*                     summary.check_packages_in_dir*
[11] summary.connection                summary.data.frame
[13] summary.Date                      summary.default
[15] summary.ecdf*                     summary.factor
[17] summary.glm                       summary.infl*
```

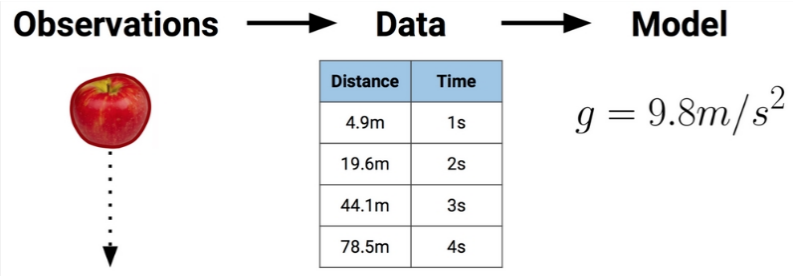
```
[19] summary.lm                              summary.loess*
[21] summary.manova                          summary.matrix
[23] summary.mlm*                            summary.nls*
[25] summary.packageStatus*                  summary.POSIXct
[27] summary.POSIXlt                         summary.powerTransform*
[29] summary.ppr*                            summary.prcomp*
[31] summary.princomp*                       summary.proc_time
[33] summary.rlang:::list_of_conditions*     summary.rlang_error*
[35] summary.rlang_message*                  summary.rlang_trace*
[37] summary.rlang_warning*                  summary.srcfile
[39] summary.srcref                          summary.stepfun
[41] summary.stl*                            summary.table
[43] summary.tukeysmooth*                    summary.vctrs_sclr*
[45] summary.vctrs_vctr*                     summary.warnings
see '?methods' for accessing help and source code
```

ML training

- Machine learning models are trained. This means that the model is fitted to a data set
- Once the model is trained, it has been transformed into an abstract form that summarizes (and transcends) the original information
- The training model is not "learning" yet because the result still must be evaluated (tested) before the model is ready.

Training a physics model

- Example from physics: by fitting equations to observational data, Newton inferred the concept of gravity (we think). It was always present but not recognized:



- In R: `g` is the acceleration due to gravity[15]

```
d <- c(4.9,19.6,44.1,78.5) # distance observations
t2 <-c(1,2,3,4)  # time observations
2*d/(t^2) # fit data to model = compute g
format(2*d/(t2^2),digits=2) # compute, print 2 digits

Error in t^2 : non-numeric argument to binary operator
[1] "9.8" "9.8" "9.8" "9.8"
```

- Other model examples include:

  1. Genomic data models identify genes responsible for disease
  2. Bank transaction models identify fraudulent activities
  3. Psychological models identify new disorders
  4. Medical models identify diagnostic patterns

- These patterns were always there but had not been identified/seen prior to presenting the information in a different format.[16]

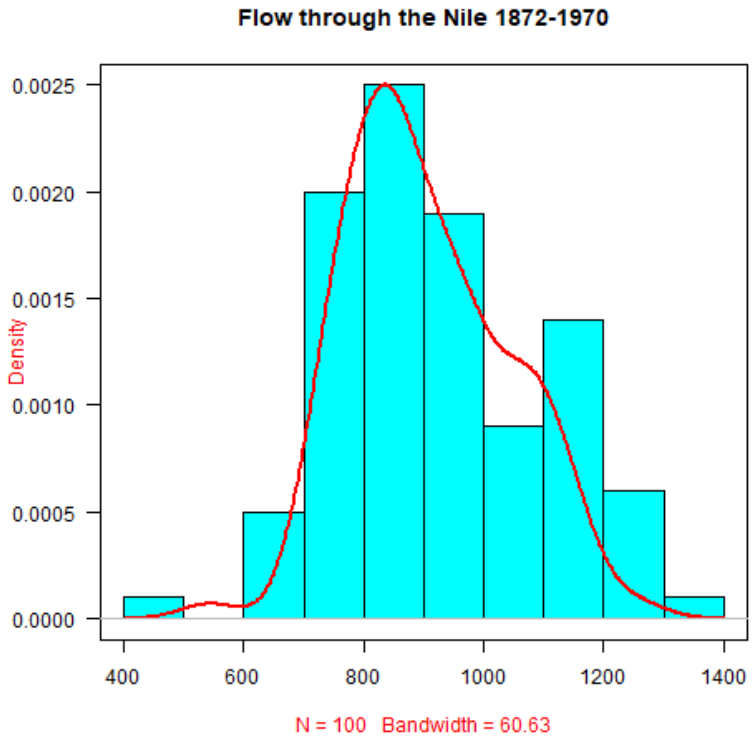
[15] For the back story on this, I asked a fully trained model, ChatGPT: "How far does an object of 1 kg fall in 1 second?" - very complete answer, check it out: tinyurl.com/mr2sm6zx.

[16] There is also the danger here - all predictions are stochastic in nature, i.e. they are probabilistic predictions, likelihoods, only. And the testing as well as the evaluation is rife with assumptions. One may ask: how permanent are the results, which are unlike gravity, subject to cultural definitions ("fraud", "disorder", "disease") hence not as objective as physical, observable laws?

Nile example - training a density model

- The `truehist` function fits the dataset to a density estimate, and `density` does the same with a smoothing effect added:

```
library(MASS)     # load MASS package
truehist(Nile,    # target dataset for histogram
 las=1,   # reorient axis labels
 xlab="", # remove default x-axis annotation
 main="") # remove default title
par(new=TRUE)     # allow plotting over previous plot
plot(density(Nile), # target dataset for plot
     col="red",   # draw line in red
     col.lab="red", # color axis label red
     lwd=2,       # double line width
     xaxt="n",    # suppress plotting x-axis
     yaxt="n",    # suppress plotting y-axis
     main="")     # remove title
title("Flow through the Nile 1872-1970")
```



Nile example: training a linear model

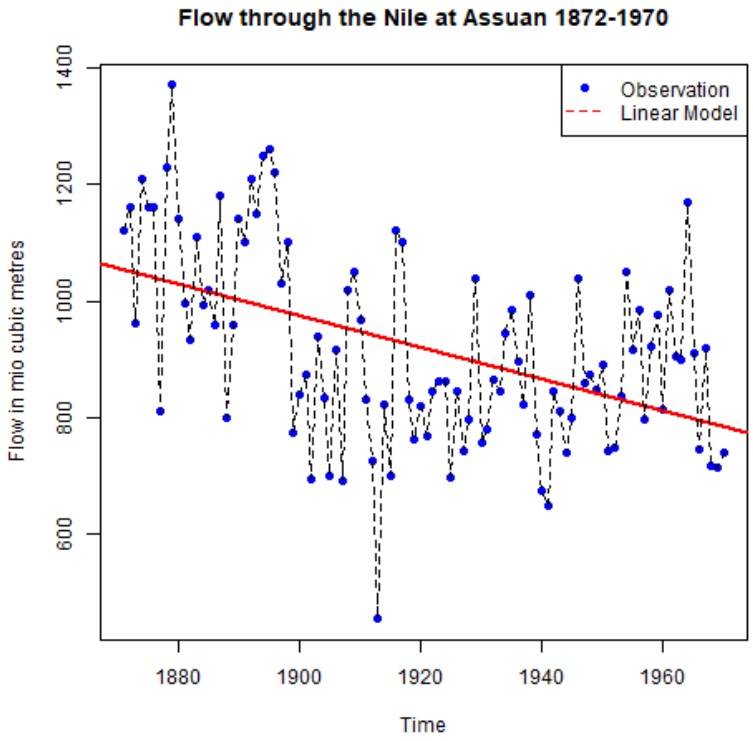
- The `lm` function needs points to fit a line through. `Nile` only has two vectors, one is the `Nile` values, the other one is the `time` of each instance of `Nile`. Apply the function `time` to `Nile`:

```
time(Nile)

Time Series:
Start = 1871
End = 1970
Frequency = 1
  [1] 1871 1872 1873 1874 1875 1876 1877 1878 1879 1880 1881 1882 1883 1884 1885
 [16] 1886 1887 1888 1889 1890 1891 1892 1893 1894 1895 1896 1897 1898 1899 1900
 [31] 1901 1902 1903 1904 1905 1906 1907 1908 1909 1910 1911 1912 1913 1914 1915
 [46] 1916 1917 1918 1919 1920 1921 1922 1923 1924 1925 1926 1927 1928 1929 1930
```

```
 [61] 1931 1932 1933 1934 1935 1936 1937 1938 1939 1940 1941 1942 1943 1944 1945
 [76] 1946 1947 1948 1949 1950 1951 1952 1953 1954 1955 1956 1957 1958 1959 1960
 [91] 1961 1962 1963 1964 1965 1966 1967 1968 1969 1970
```

- The `lm` function attempts to fit a linear model to the `Nile` dataset:

```
## create the linear model (needs 2 dimensions)
model <- lm(Nile ~ time(Nile))

## plot Nile data
plot(Nile,
     type="p",  # plot points only
     col="blue", # plot points in blue
     pch=16,  # point character solid circle
     ylab="Flow in mio cubic metres") # y-axis annotation

## draw the model - a trendline
abline(model,  # model consists of intercept and slope
       col="red", # red line
       lwd=2)   # double line width

## connect Nile data by black dashed lines
lines(Nile,
      type="l",
      col="black",
      lty=2)

## title plot
title("Flow through the Nile at Assuan 1872-1970")

## add a legend
legend("topright",  # where the legend is located
       legend=c("Observation", "Linear Model"),
       pch = c(16,NA),  # assign point character
       lty = c(NA, 2),  # assign line type
       col = c("blue", "red")) # assign color
```



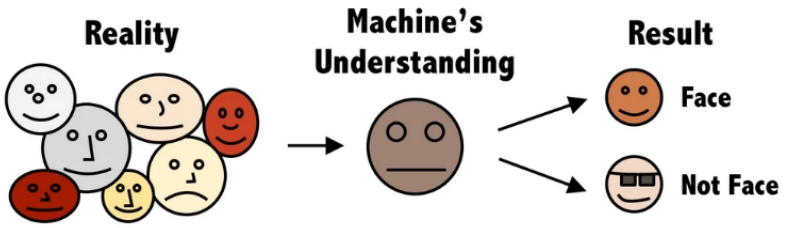
- This last example demonstrates "underfitting" = most points are not well represented by the model. However, the general trend is well represented by the red line: over time, the water flow through the Nile at Assuan decreased.

Generalization

- **Generalization** uses abstracted data to create knowledge and inferences that drive action in new contexts
- To do this, the machine searches through an entire set of models (equivalent to theories of prediction or inference) employing a process called "heuristics" (finding skills or educated guesses)
- Compare it to a Google search that you perform yourself: in response to the output of the search you refine your search string, e.g.

1. "generalization" (in response to the too general result)
2. "generalization reasoning" (in response to Google's completion)
3. "generalization reasoning models" (in response to your interest)
4. "generalization models" (in response to the too specific result)
5. "generalization machine learning" (result still too specific)

- **Human heuristics** are guided by emotion and can be fallible - e.g. "availability heuristics", the tendency to estimate likelihood of an event depending on how easily examples can be recalled (e.g. airline accidents over vehicle accidents)
- Misapplied **machine heuristics** as a result of algorithmic errors are called **bias** if the conclusions are *systematically* erroneous (i.e. wrong in a consistent or predictable manner)
- Example: an ML algorithm that generalizes faces to have two circles above a mouth would not identify a face with glasses.



☐ Could "a little bias" also be useful?[17]

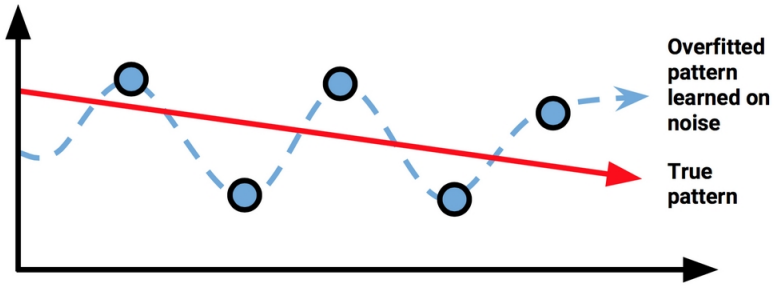
[17] Bias (like presets) allows us to favor some choices over others and discard some choices as irrelevant. The net effect could be that we become more action-oriented and less bogged down by search. Bias is also how ML algorithms choose among many ways to understand data.

Evaluation + overfitting

"There is no single learning algorithm to rule them all." -Brett Lantz

- No ML approach is best for every problem - an application of the rigorous "No Free Lunch" (NFL) theorem for search and optimization (Wolpert/Macready, 2005)
- **Evaluation** provides a feedback mechanism to measure the utility of learned knowledge and inform potential improvements
- After training on an initial training dataset, the model is evaluated on a separate test dataset of new, unseen cases
- Models fail to generalize perfectly due to noise, unexplained or inexplicable variations in data due to
  1. **measurement** errors (e.g. imprecise sensors)
  2. **human** subject issues (e.g. random answers in surveys)
  3. data **quality** issues (missing, null, truncated, corrupted values)
  4. **complex** phenomena whose impact appears to be random
- Famous noise that turned into gold: cosmic microwave background radiation that is attributed to an echo of the 'Big Bang'
- Modeling noise is called *overfitting*.[18]

[18] Compare this with the very similar looking diagram earlier, the linear trendline modeling of the `Nile` time series data. Underfitting misses out on available information, while overfitting interprets noise (irrelevant information) as meaningful for the pattern.



Summary

- ML can find actionable insight in large data sets
- ML involves **abstraction** of data into structured **representation** and **generalization** of the structure into action that can be **evaluated**
- Data that contains examples/observations/records and features of the concept to be learnt is summarized in a **model**

ML Glossary

TERM	MEANING
Machine learning	Computer solves task with models
Data abstraction	Transform raw data to table structure
Generalization	Trained model fits unknown data
Evaluation	Model feedback to test accuracy
Heuristics	Model to find solution quickly
Bias	Machine heuristics that lead to errors
Overfitting	Modeling noise instead of signals
Underfitting	Model is too simple for the data

References

- Anderson (2017). Twenty years on from Deep Blue vs Kasparov: how a chess match started the big data revolution. @theconversation.com.
- Hosseini, Z., Hytönen, K., & Kinnunen, J. (2022). Improving Online Content Quality Through Technological Pedagogical Content Design (TPCD). In S. Vachkova, & S. S. Chiang (Eds.), Education and City:

Quality Education for Modern Cities, vol 3. European Proceedings of Educational Sciences (pp. 284-296). European Publisher. https://doi.org/10.15405/epes.22043.25

- Lantz (2019). Machine Learning with R. Packt.
- Roiger (2020). Just Enough R!. CRC Press.
- Serrano (2021). Grokking Machine Learning.