Lecture 01

Data and model types

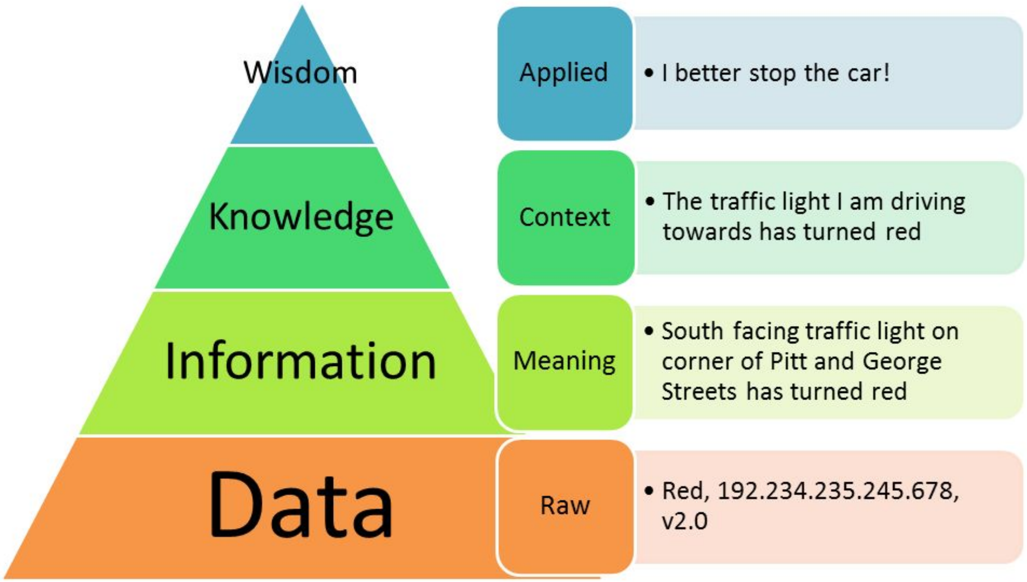
https://github.com/dalcimar/
UTFPR - Federal University of Technology - Paraná
https://www.dalcimar.com/

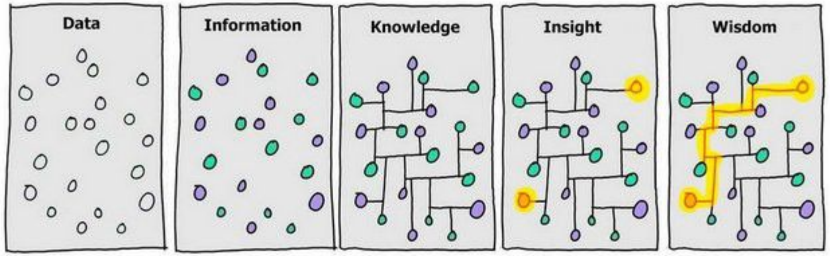
O que é informação?

Informação é a **resultante do processamento**, **manipulação** e **organização de dados**, de tal forma que represente uma **modificação** (quantitativa ou qualitativa) no **conhecimento do sistema** (humano, animal ou máquina) que a recebe.

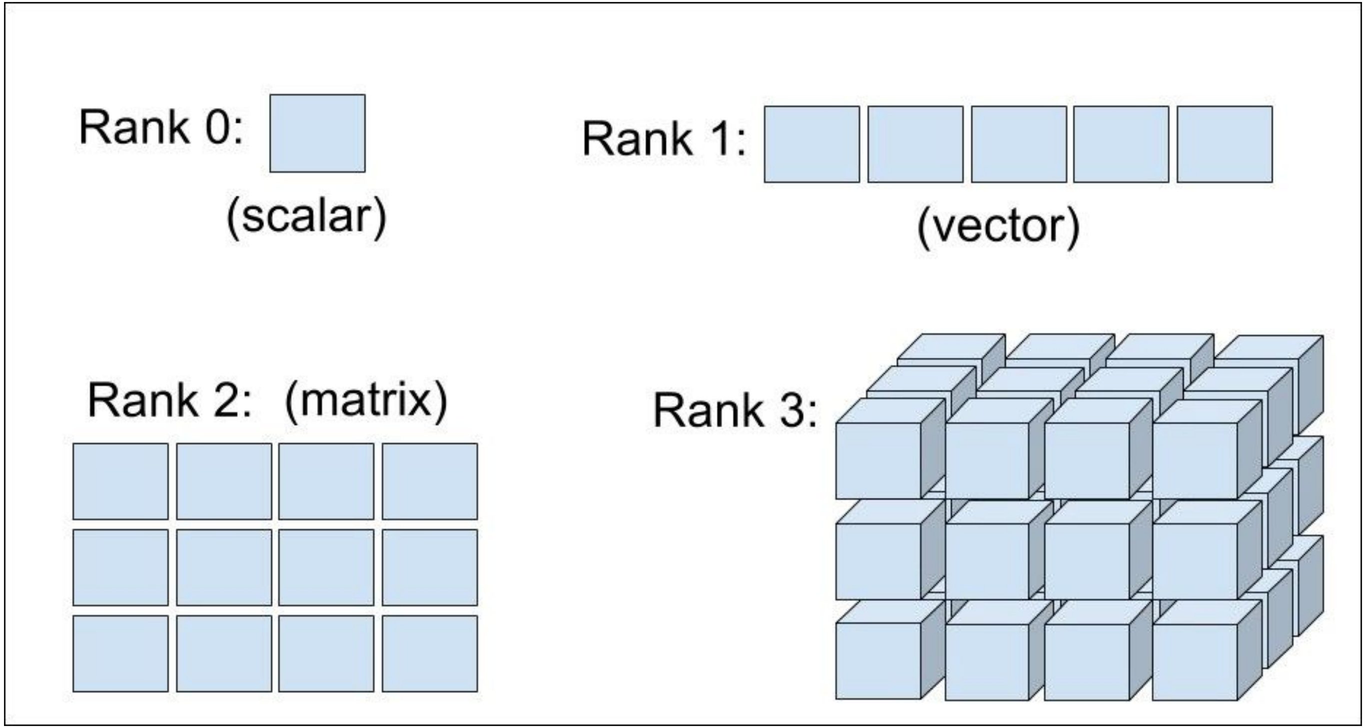
DIKW model

Data-Information-Knowledge-Wisdom

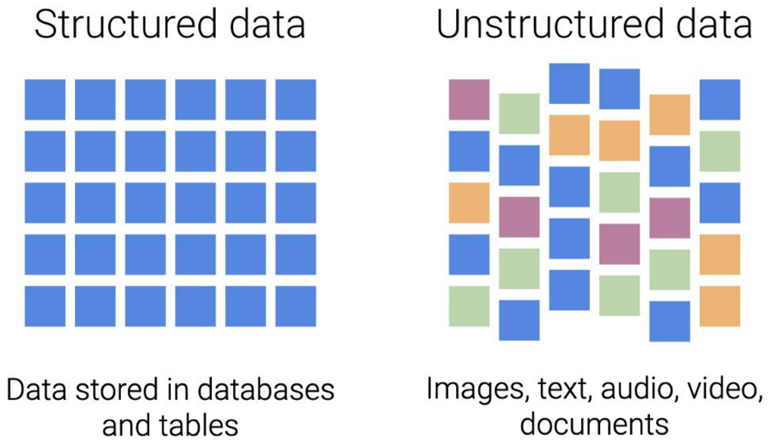




Data format



Data types



Structured data resides in relational databases: a database structured to recognise relations between stored items of data

- structured query language ("SQL") to access and manipulate items

Unstructured data is everything else.

- unstructured data has an internal structure (i.e. bits and bytes)
- but is not structured via pre-defined data models or schema, i.e. not organised and labelled to identify meaningful relationships between data
- it may be textual / non-textual. It may be human / machine-generated. It might also be stored within a non-relational database like NoSQL.

Structured Data vs Unstructured Data

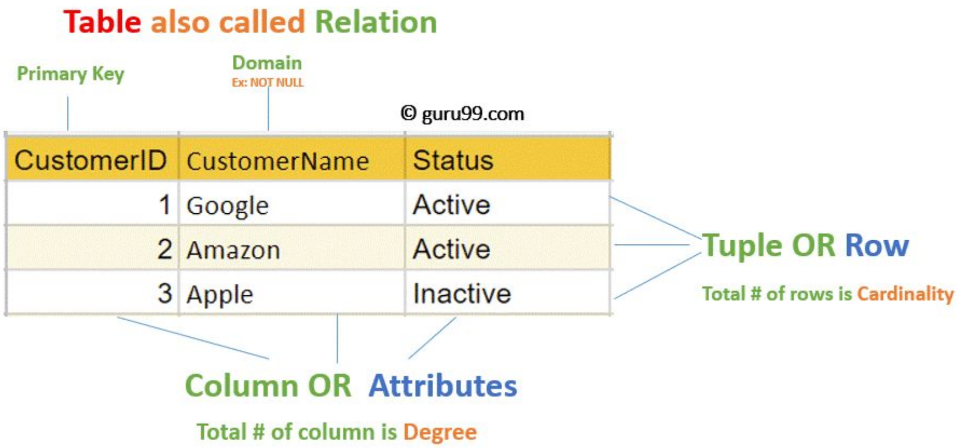
Structured Data	Unstructured Data
Can be displayed in rows, columns and relational databases	Cannot be displayed in rows, columns and relational databases
Numbers, dates and strings	Images, audio, video, word processing files, e-mails, spreadsheets
Estimated 20% of enterprise data *(Gartner)*	Estimated 80% of enterprise data *(Gartner)*
Requires less storage	Requires more storage
Easier to manage and protect with legacy solutions	More difficult to manage and protect with legacy solutions

Data types

	Structured Data	Unstructured Data
Characteristics	• Pre-defined data models • Usually text only • Easy to search	• No pre-defined data model • Can be text, images, sound, video and other formats • Difficult to search
Resides in	• Relational databases • Data warehouses	• Applications • NoSQL databases • Data warehouses • Data lakes • Cloud folders
Examples	• Dates • Phone numbers • Social security numbers • Credit card numbers • Customer names • Addresses • Product names and numbers • Transaction information	• Text files • Reports • Email messages • Audio files • Video files • Images • Surveillance imagery

Structured data

Tables and relational databases



CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Inactive

Customer

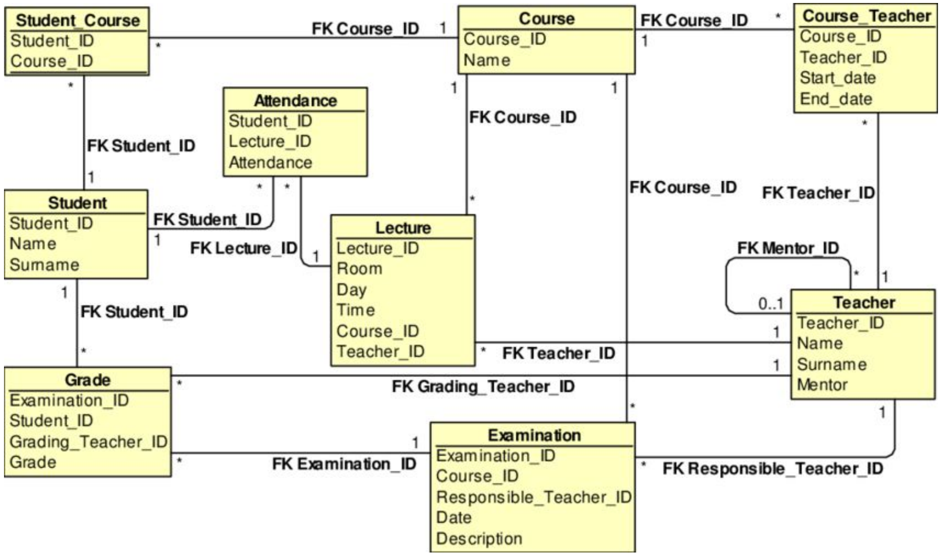
Billing

InvoiceNo	CustomerID	Amount
1	1	$100
2	1	$200
3	2	$150

Structured data

Play Tennis ?????

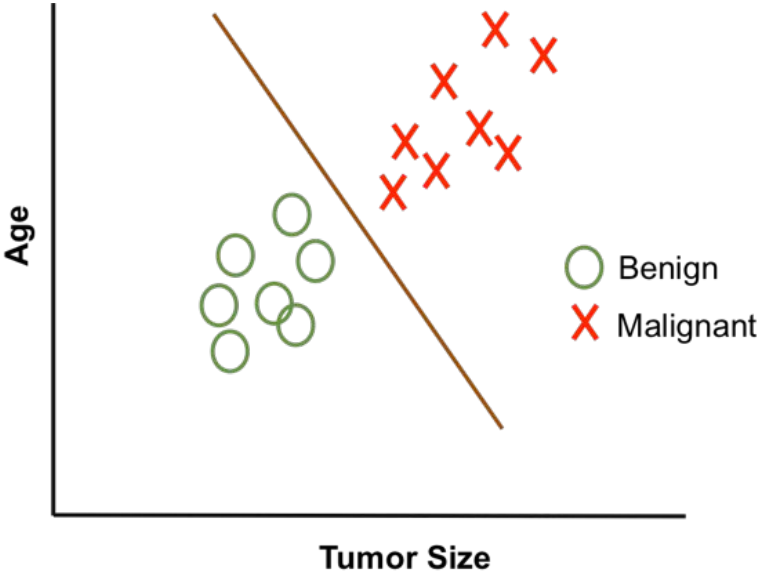
Day	Outlook	Temperature	Humidity	Wind	PlayTennis
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	overcast	hot	high	weak	yes
4	rain	mild	high	weak	yes
5	rain	cool	normal	weak	yes
6	rain	cool	normal	strong	no
7	overcast	cool	normal	strong	yes
8	sunny	mild	high	weak	no
9	sunny	cool	normal	weak	yes
10	rain	mild	normal	weak	yes
11	sunny	mild	normal	strong	yes
12	overcast	mild	high	strong	yes
13	overcast	hot	normal	weak	yes
14	rain	mild	high	strong	no



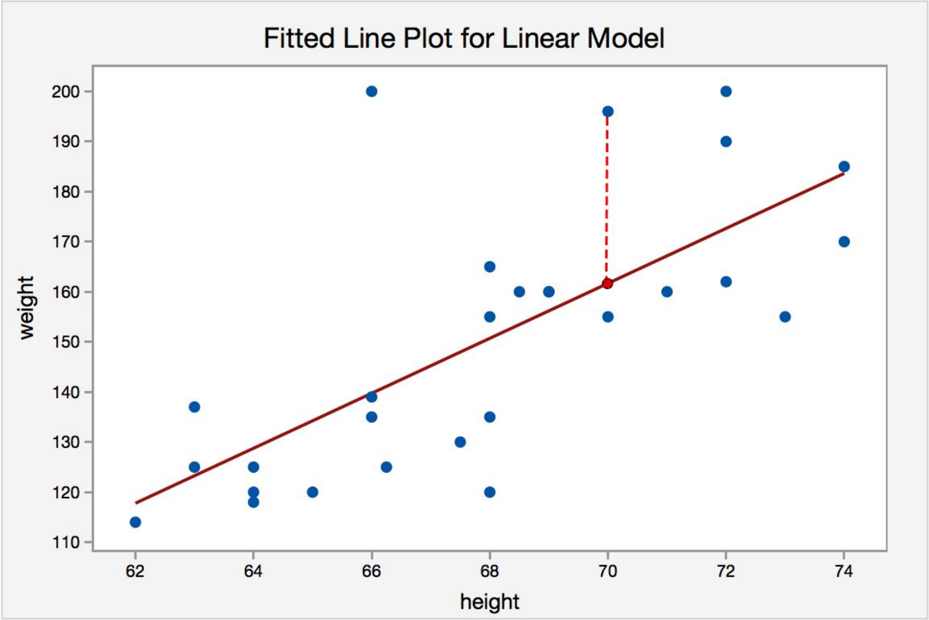
Obs	HospitalID	Sex	Cholesterol	BP_Status
1	2	Male	194	Normal
2	3	Female	200	High
3	0	Male	233	High
4	1	Female	192	Optimal
5	2	Female	209	Normal
6	3	Female	200	High
7	0	Female	184	Normal
8	1	Female	228	High
9	2	Female	150	Normal
10	3	Male	221	Normal

Structured data tasks

Classification

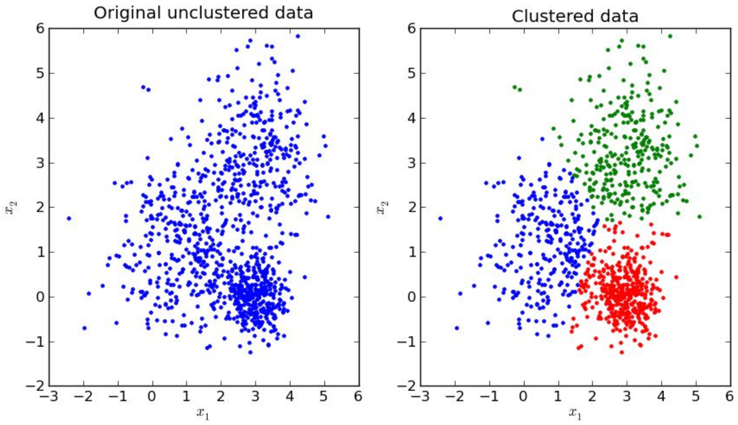


Regression

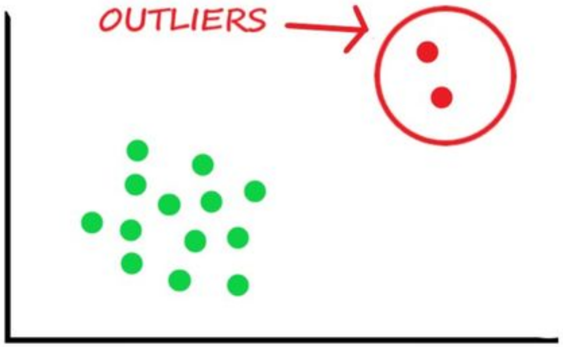


Structured data tasks

Clustering



Outliers detection



Structured data tasks

Some unstructured data can be turned on structured data with loss of information

	MARY	IS	HUNGRY	HAPPY	FOR	APPLES	NOT	JOHN	HE	
"Mary is hungry for apples." →	1	1	1	0	1	1	0	0	0	→ [1, 1, 1, 0, 1, 1, 0, 0, 0]
"John is happy he is not hungry for apples." →	0	2	1	1	1	1	1	1	1	→ [0, 2, 1, 1, 1, 1, 1, 1, 1]

Structured data tasks

Association mining

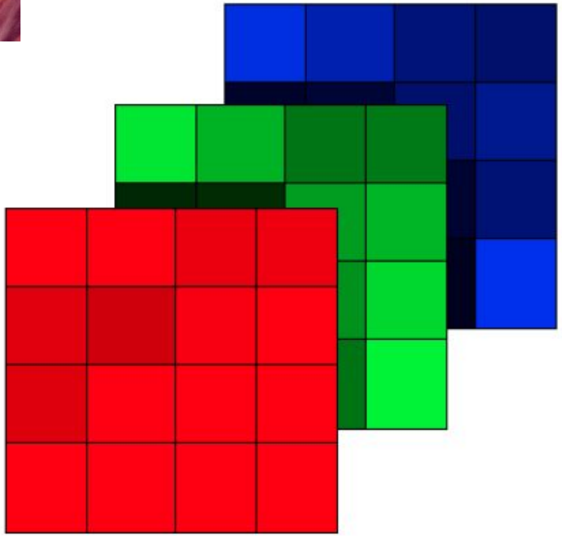
ID	Items
1	{Bread, Milk}
2	{Bread, Diapers, Beer, Eggs}
3	{Milk, Diapers, Beer, Cola}
4	{Bread, Milk, Diapers, Beer}
5	{Bread, Milk, Diapers, Cola}
…	…

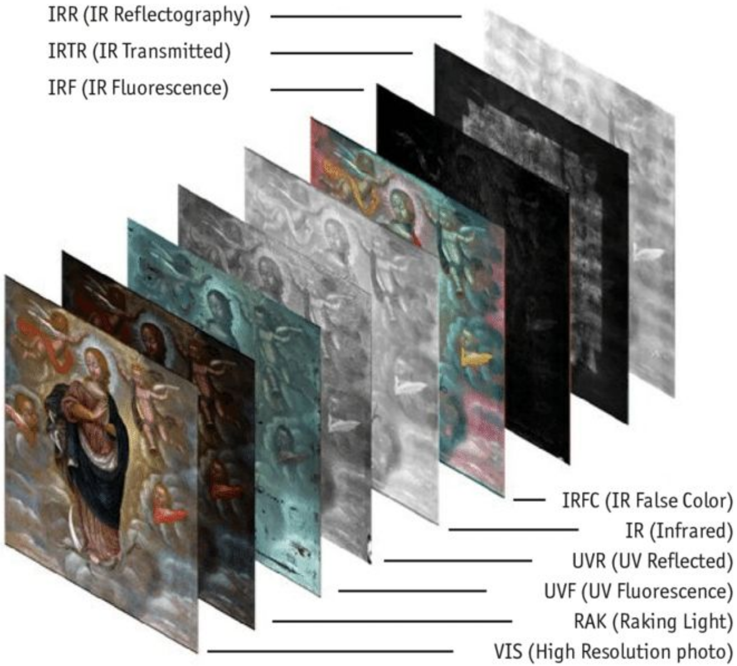
{Diapers} → {Beer} Example of an association rule

	shrimp	almonds	avocado	vegetables mix	green grapes	whole weat flour	yams	cottage cheese	energy drink	tomato juice	low fat yogurt	green tea	honey	salad	mineral water	salmon	antioxydant juice
0	burgers	meatballs	eggs	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	chutney	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	turkey	avocado	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	mineral water	milk	energy bar	whole wheat rice	green tea	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	low fat yogurt	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

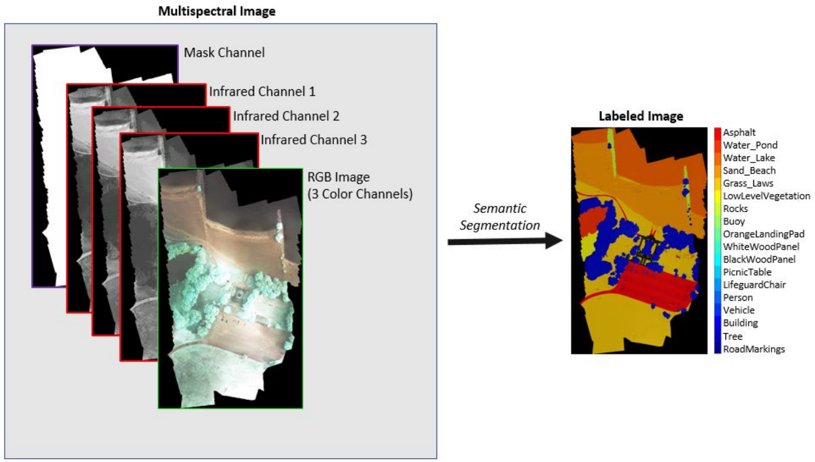
Unstructured data (2D signals)

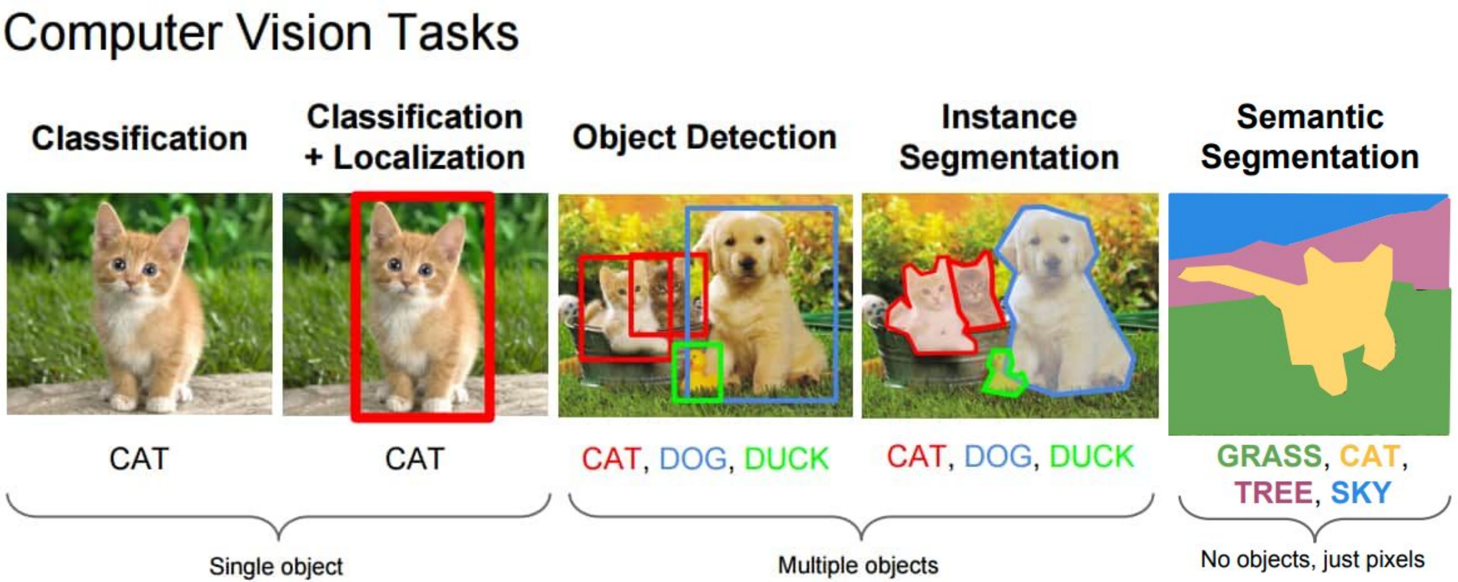
Unstructured data (3D and ND signals)



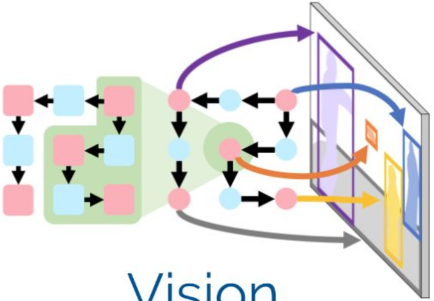


Images tasks



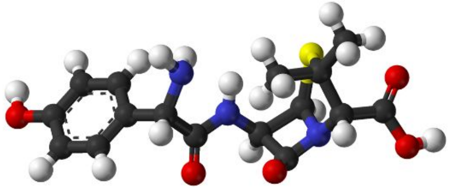


Unstructured data (graphs)

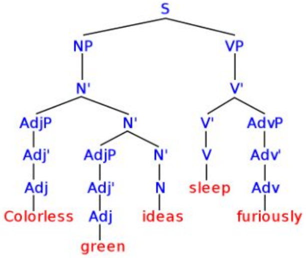


Vision

Knowledge graphs

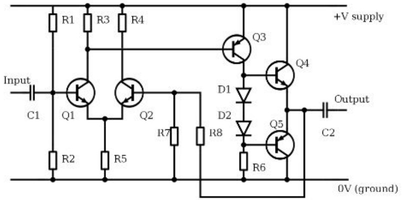


Chemistry



Language

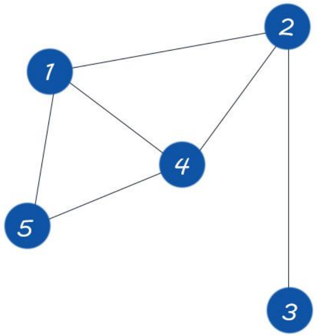
Traffic networks



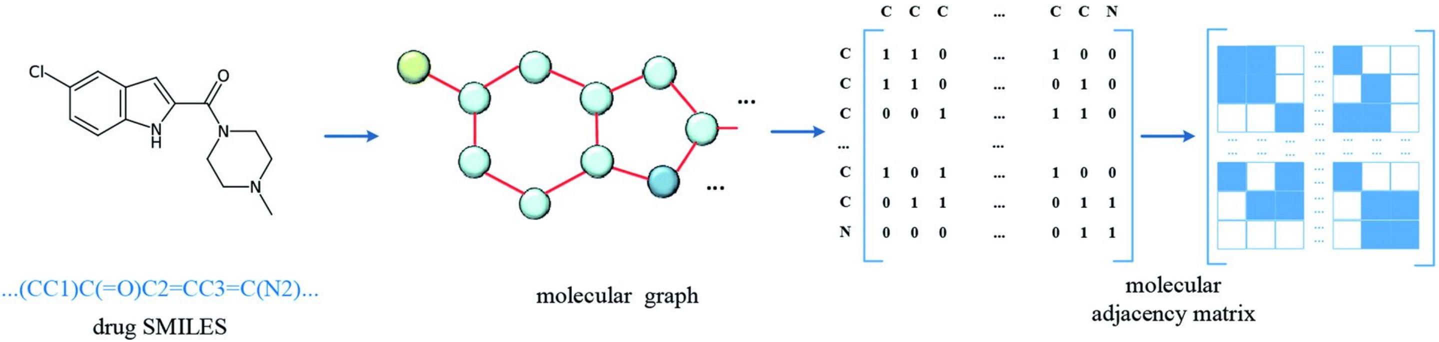
Circuits

Social networks

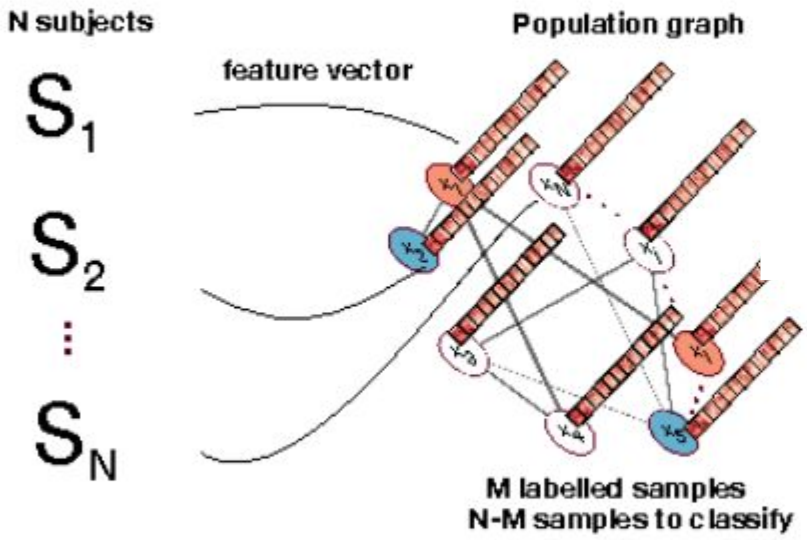
Unstructured data (graphs)

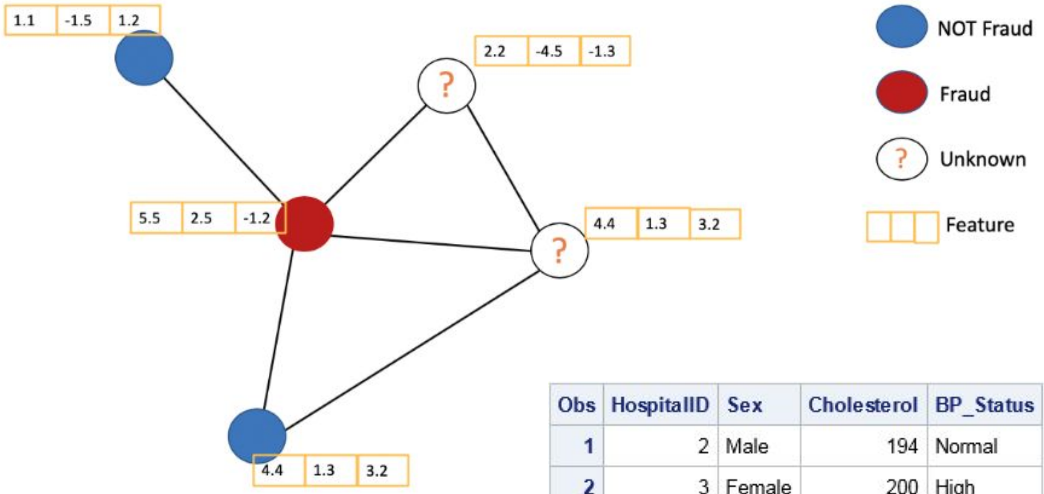


$$A = \begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 0 & 1 & 0 & 1 & 1 \\ 2 & 1 & 0 & 1 & 1 & 0 \\ 3 & 0 & 1 & 0 & 0 & 0 \\ 4 & 1 & 1 & 0 & 0 & 1 \\ 5 & 1 & 0 & 0 & 1 & 0 \end{array}$$



Unstructured data (graphs)



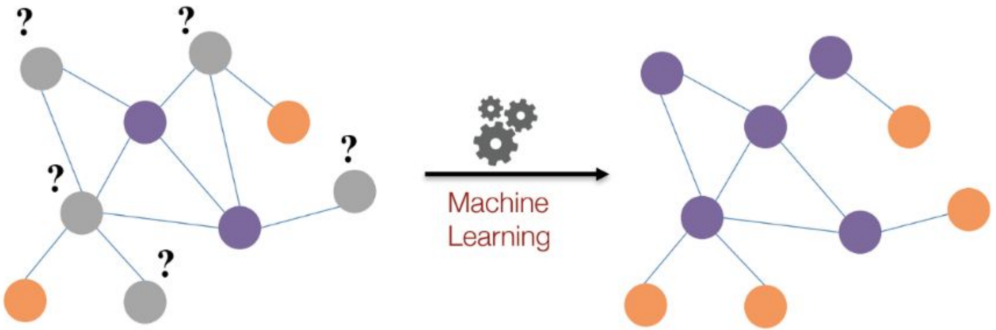


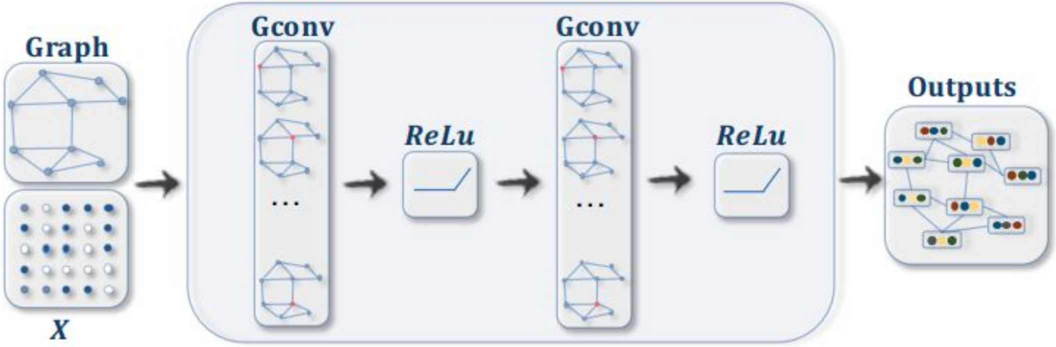
Obs	HospitalID	Sex	Cholesterol	BP_Status
1	2	Male	194	Normal
2	3	Female	200	High
3	0	Male	233	High
4	1	Female	192	Optimal
5	2	Female	209	Normal
6	3	Female	200	High
7	0	Female	184	Normal
8	1	Female	228	High
9	2	Female	150	Normal
10	3	Male	221	Normal

Graph tasks

Node-level outputs relate to node regression and node classification tasks.

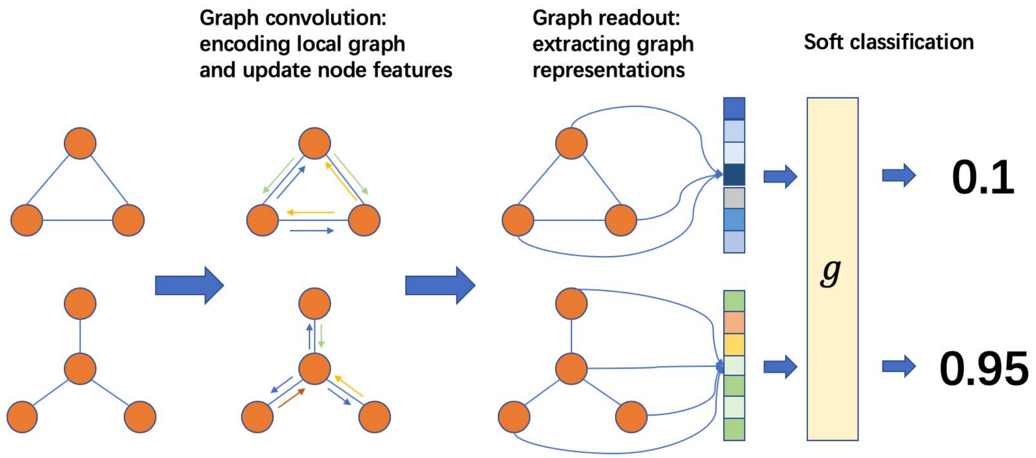
Edge-level outputs relate to the edge classification and link prediction tasks.

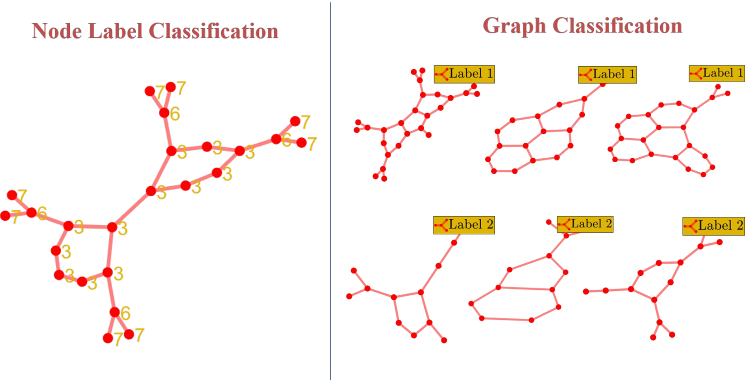


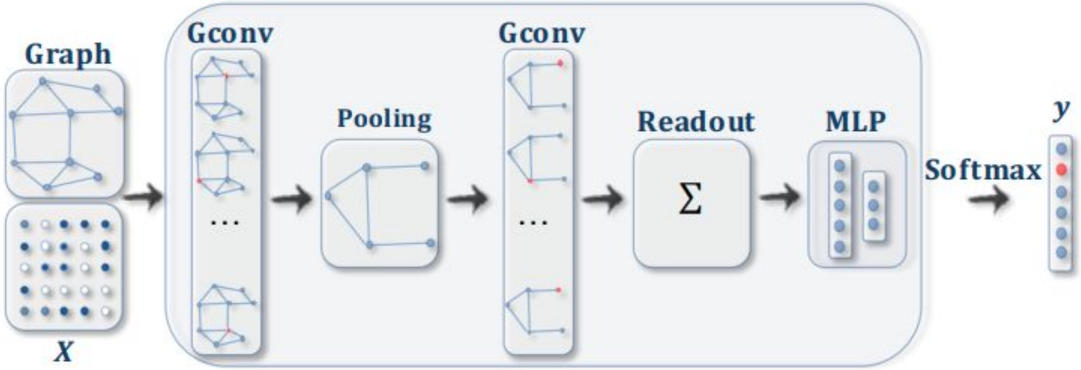


Graph tasks

Graph-level outputs relate to the graph classification task

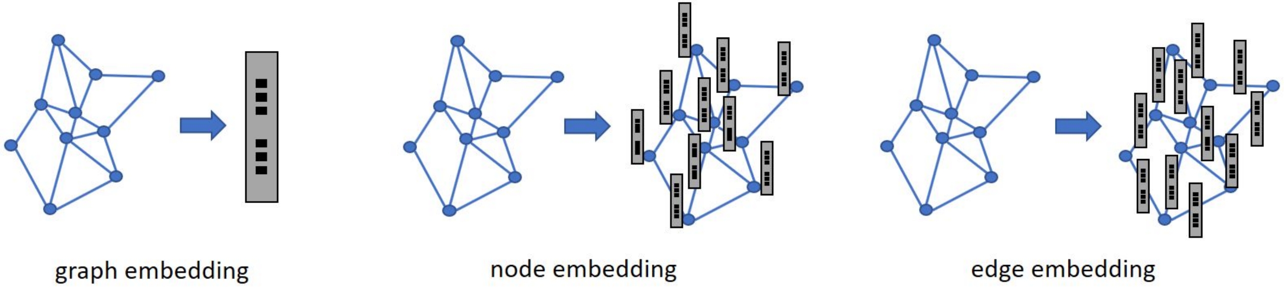


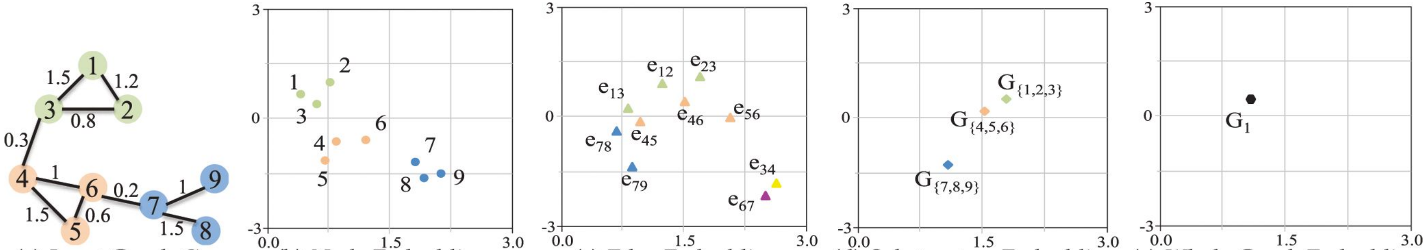




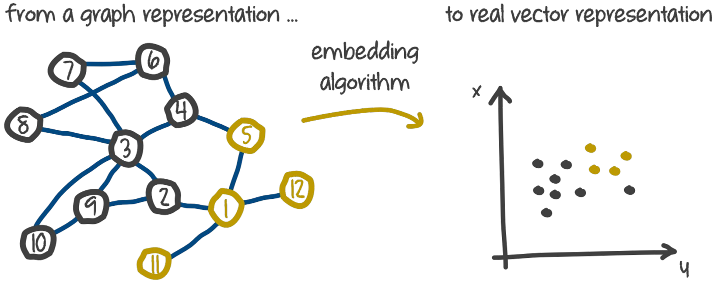
Graph tasks

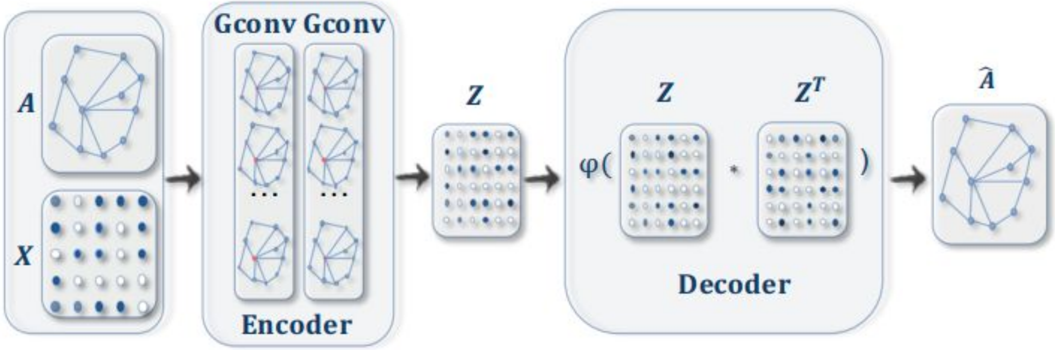
Representation learning





(a) Input Graph G_1 (b) Node Embedding (c) Edge Embedding (d) Substructure Embedding (e) Whole-Graph Embedding





Applications:
Working with Sequential Data

- Text classification
- Speech recognition (acoustic modeling)
- language translation
- ...

Stock market predictions

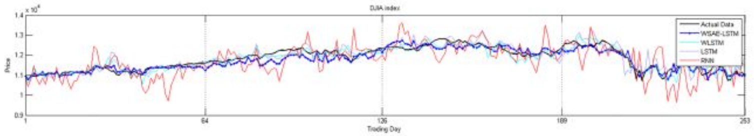
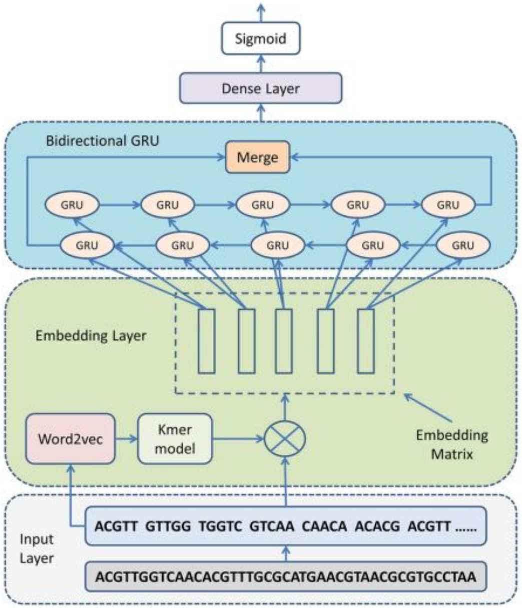


Fig 8. Displays the actual data and the predicted data from the four models for each stock index in Year 1 from 2010.10.01 to 2011.09.30.

https://doi.org/10.1371/journal.pone.0180944.g008

Bao, Wei, Jun Yue, and Yulei Rao. "[A deep learning framework for financial time series using stacked autoencoders and long-short term memory.]" *PloS one* 12, no. 7 (2017): e0180944.



Shen, Zhen, Wenzheng Bao, and De-Shuang Huang. "[Recurrent Neural Network for Predicting Transcription Factor Binding Sites.]" *Scientific reports* 8, no. 1 (2018): 15270.

DNA or (amino acid/protein) sequence modeling

Different Types of Sequence Modeling Tasks

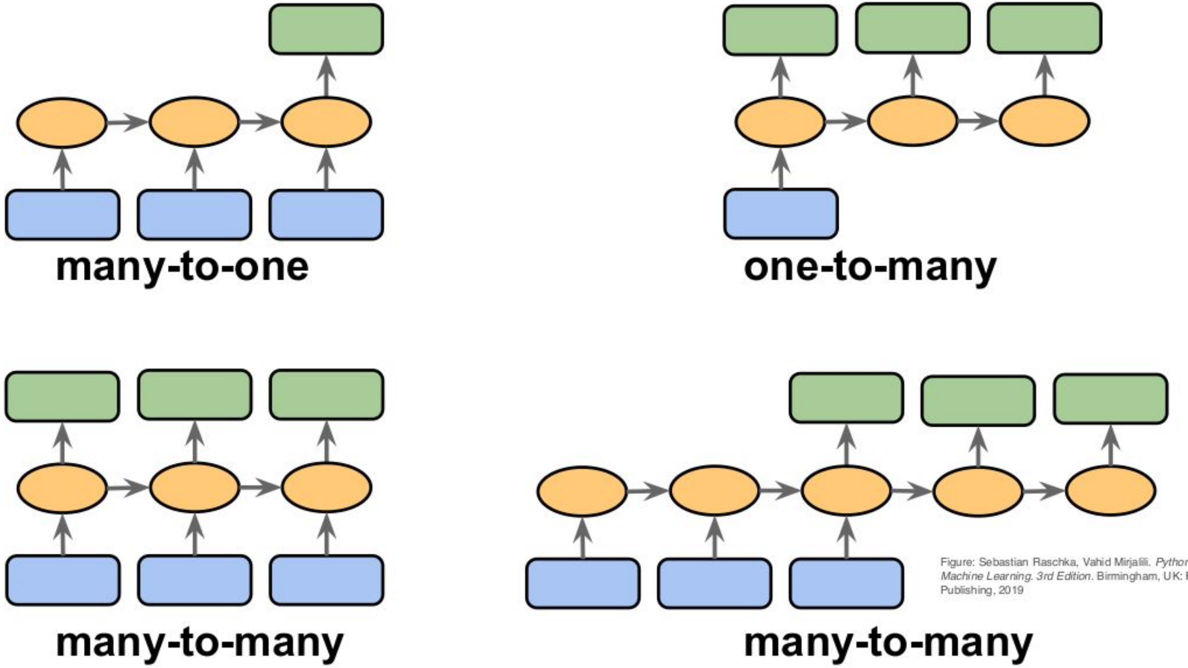
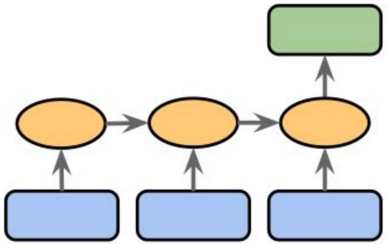


Figure: Sebastian Raschka, Vahid Mirjalili. *Python Machine Learning. 3rd Edition*. Birmingham, UK: Packt Publishing, 2019

Figure based on:

The Unreasonable Effectiveness of Recurrent Neural Networks by Andrej Karpathy (http://karpathy.github.io/2015/05/21/rnn-effectiveness/)

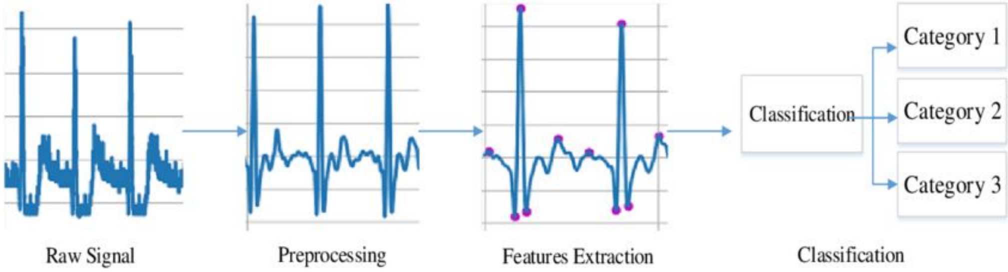
Different Types of Sequence Modeling Tasks

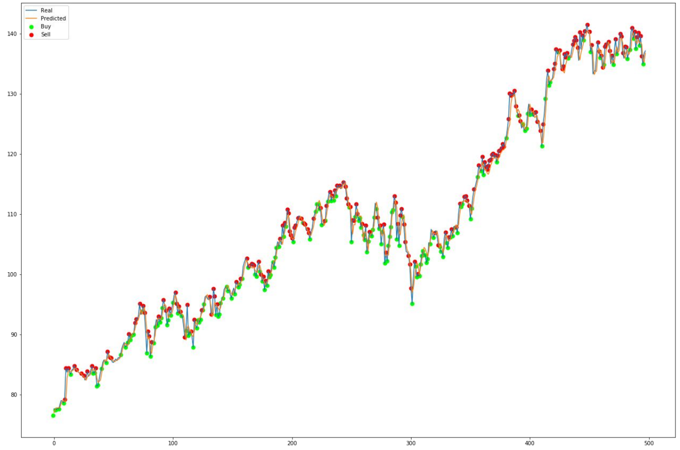


Many-to-one: The input data is a sequence, but the output is a fixed-size vector, not a sequence.

Ex.: sentiment analysis, the input is some text, and the output is a class label.

Sequence tasks on 1d signals (many to one)

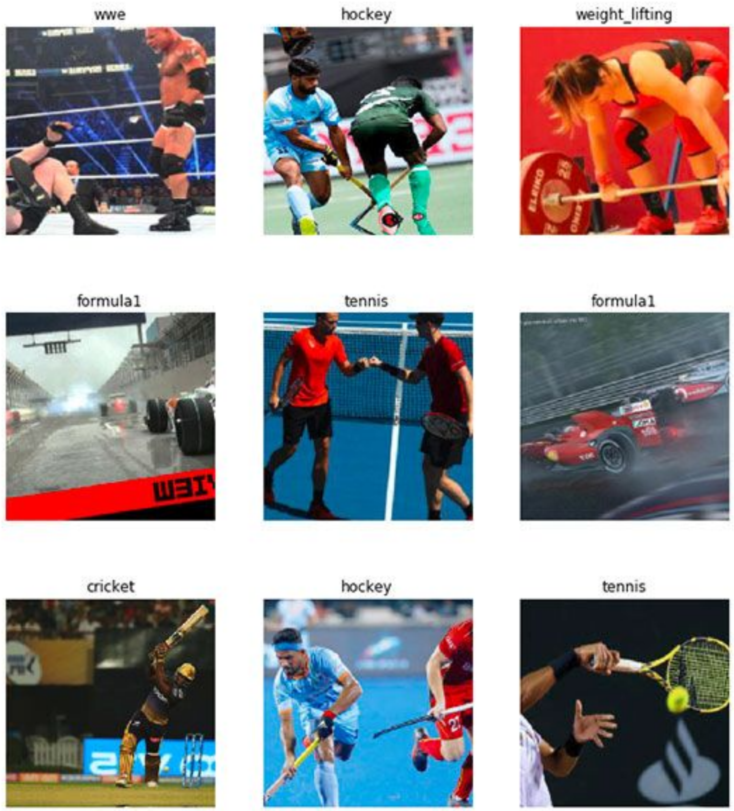




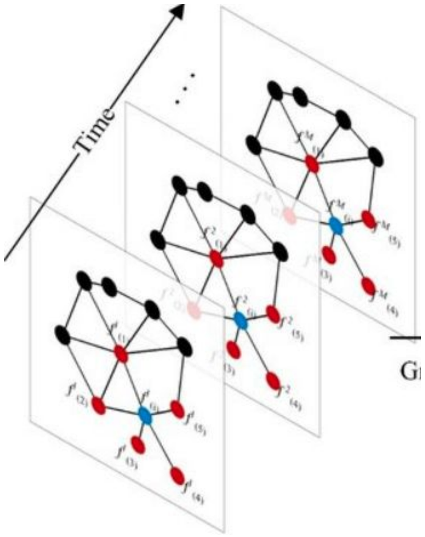
Sequence tasks on 2d and ND signals (many to one)

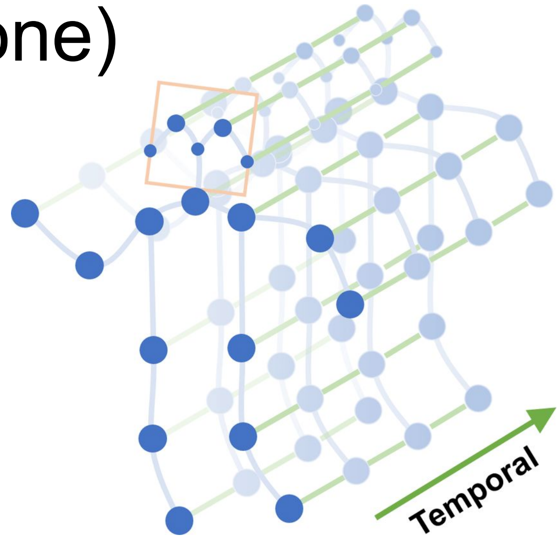
[Re3: Real-Time Recurrent Regression Networks for Visual Tracking of Generic Objects]()

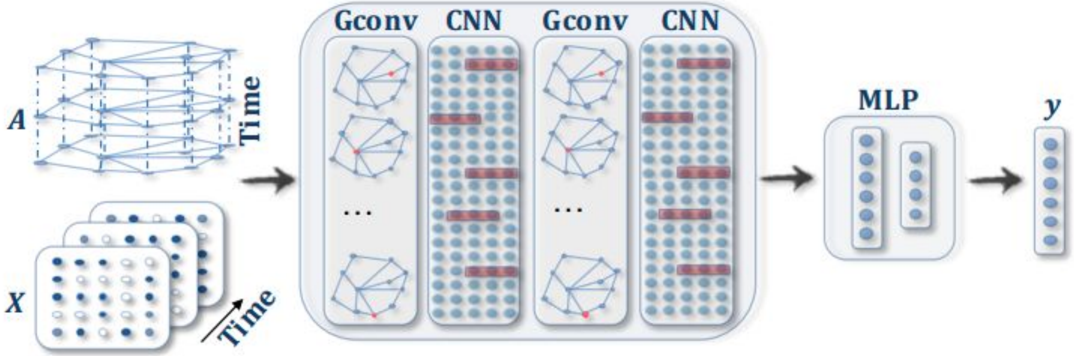
[LSTM based object tracker | Computer vision | Tracking | TorqueVision]()



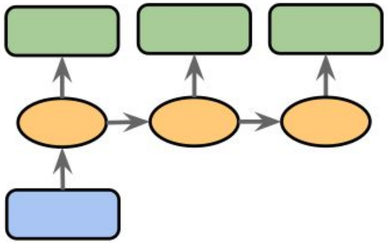
Sequence tasks on graphs (many to one)







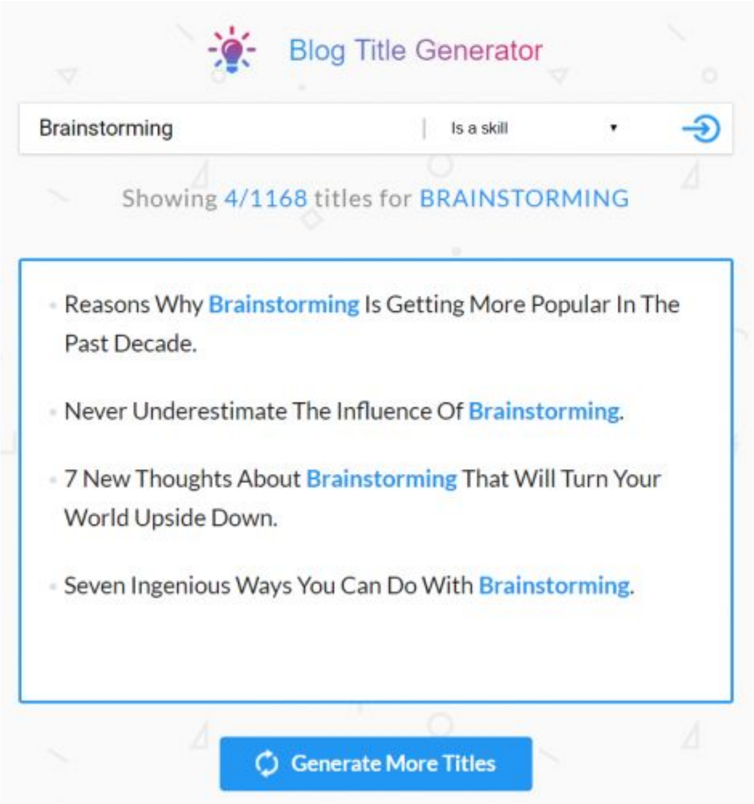
Different Types of Sequence Modeling Tasks



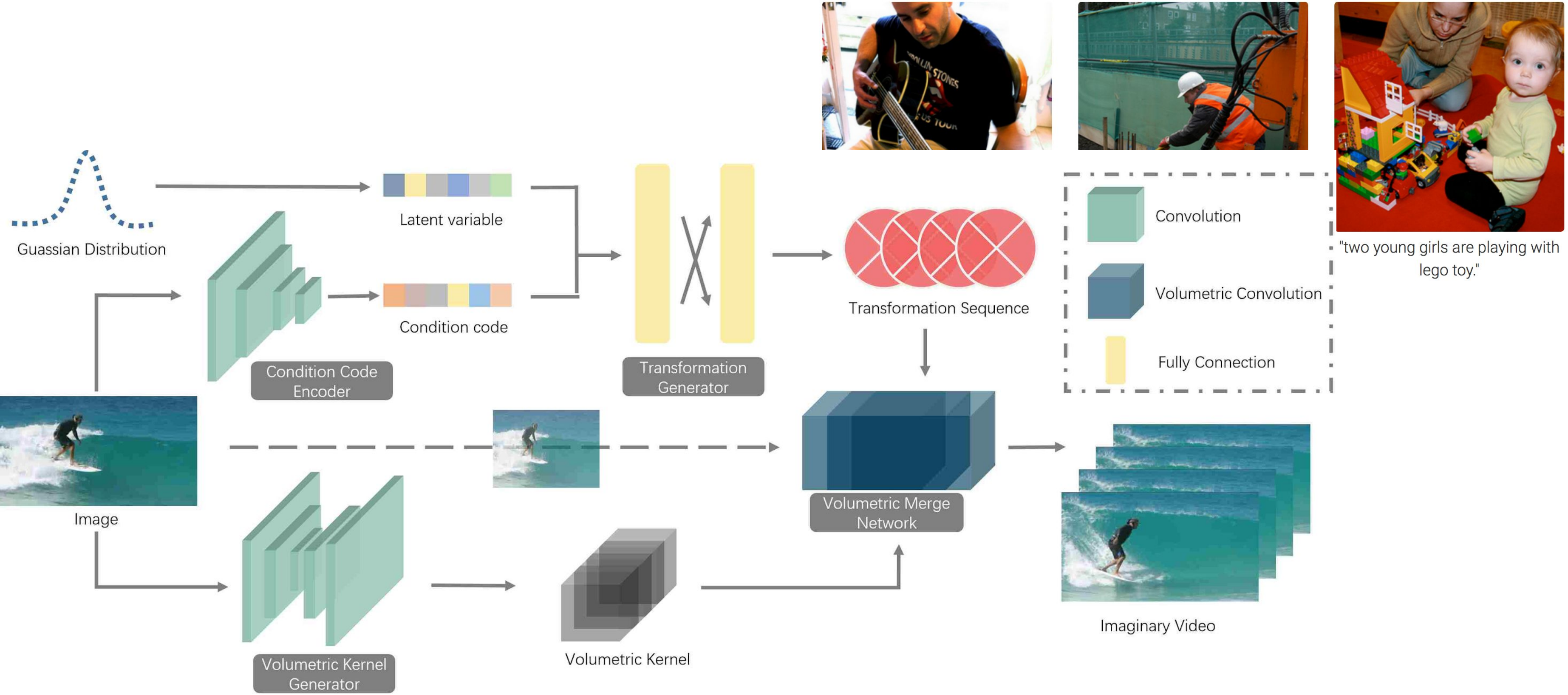
One-to-many: Input data is in a standard format (not a sequence), the output is a sequence.

Ex.: Image captioning, where the input is an image, the output is a text description of that image

Sequence tasks on 1D signals (one to many)



Sequence tasks on 2d and ND signals (one to many)

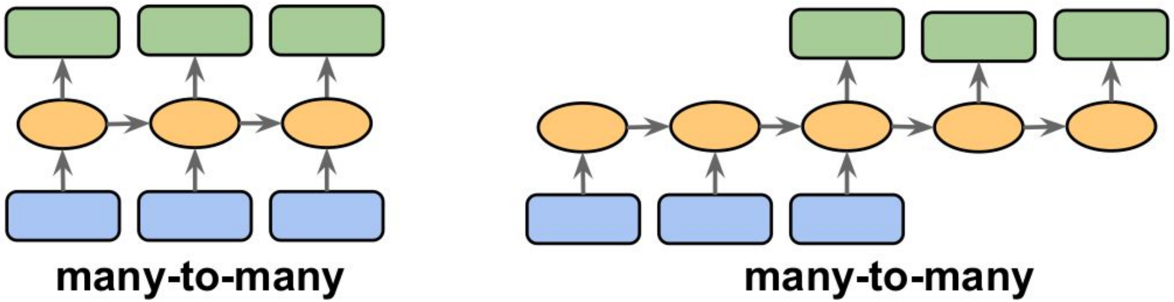


Different Types of Sequence Modeling Tasks

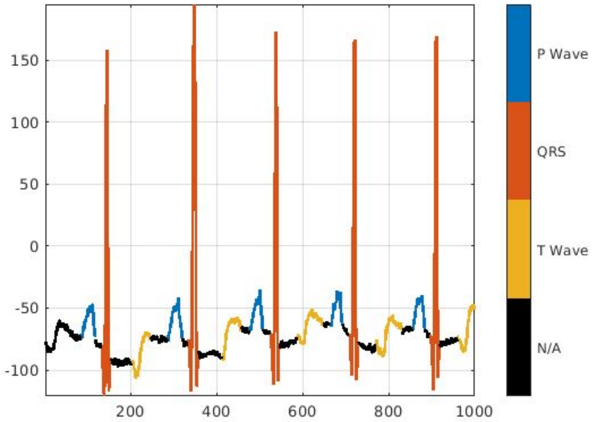
Many-to-many: Both inputs and outputs are sequences. Can be direct or delayed.

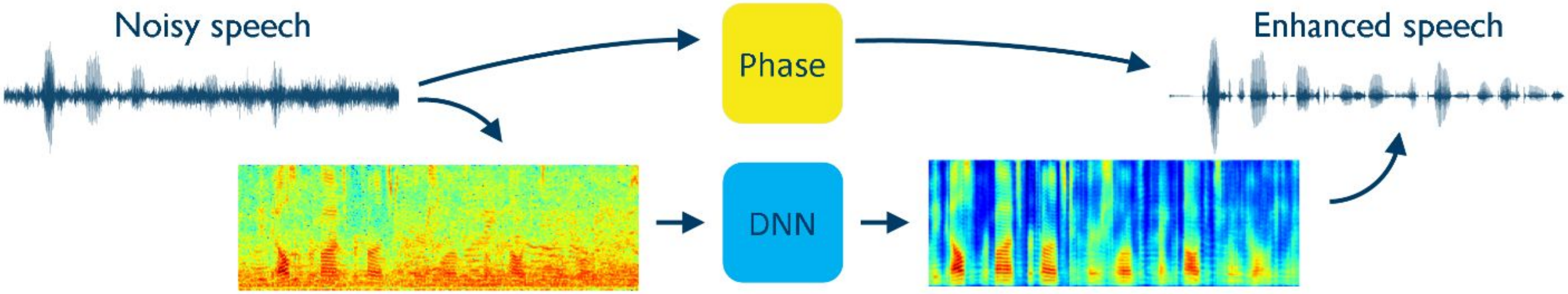
Ex.: Video-captioning, i.e., describing a sequence of images via text (direct).

Translating one language into another (delayed)



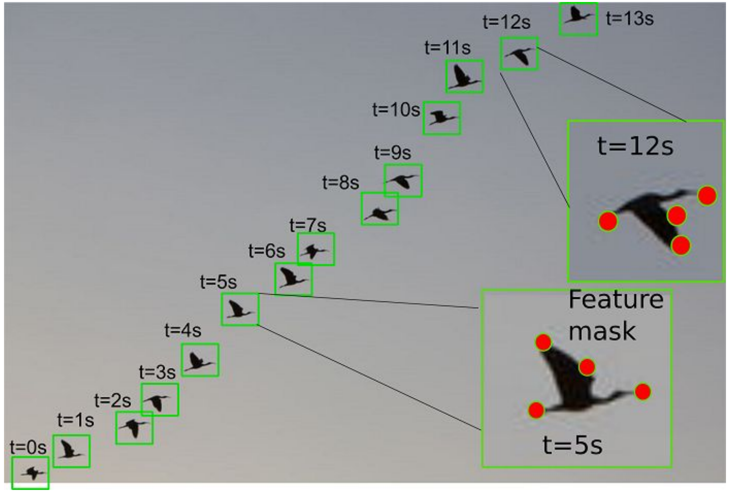
Sequence tasks on 1D signals (many to many)

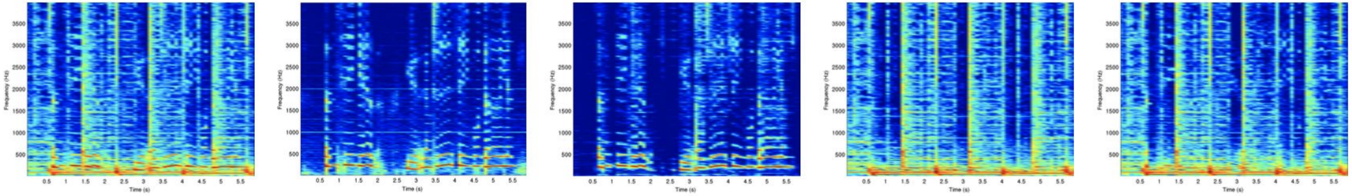




Sequence tasks on ND signals (many to many)

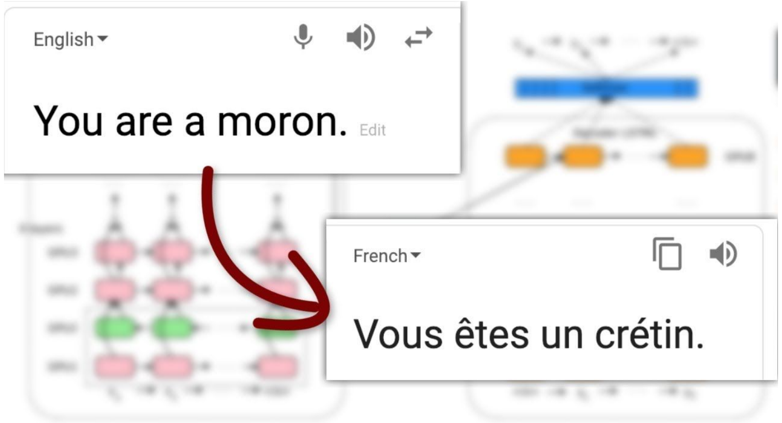
ICNet for Real-Time Semantic Segmentation on High-Resolution Images

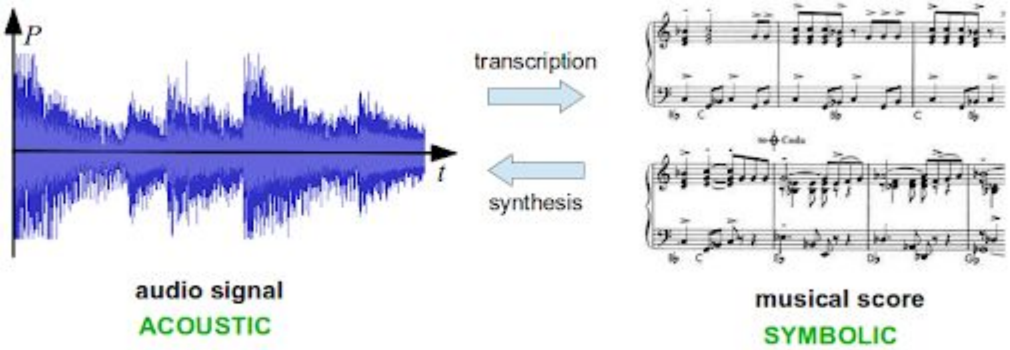


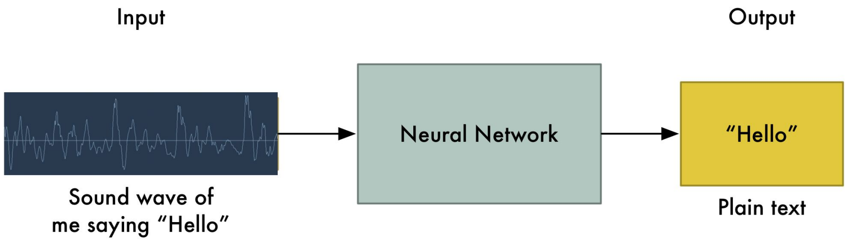


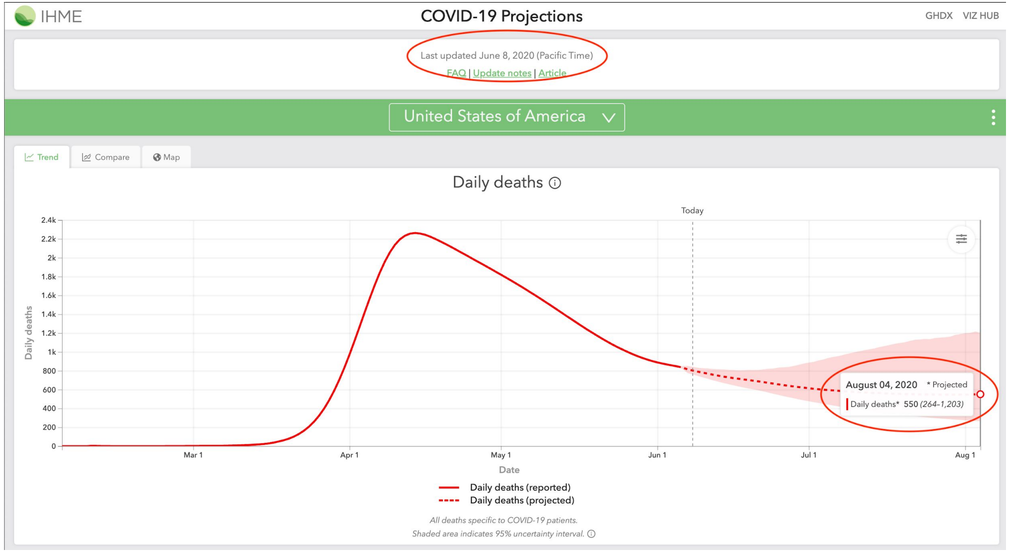
(a) Mixutre (b) Clean vocal (c) Recovered vocal (d) Clean music (e) Recovered music

Sequence tasks on 1D signals (many to many delayed)

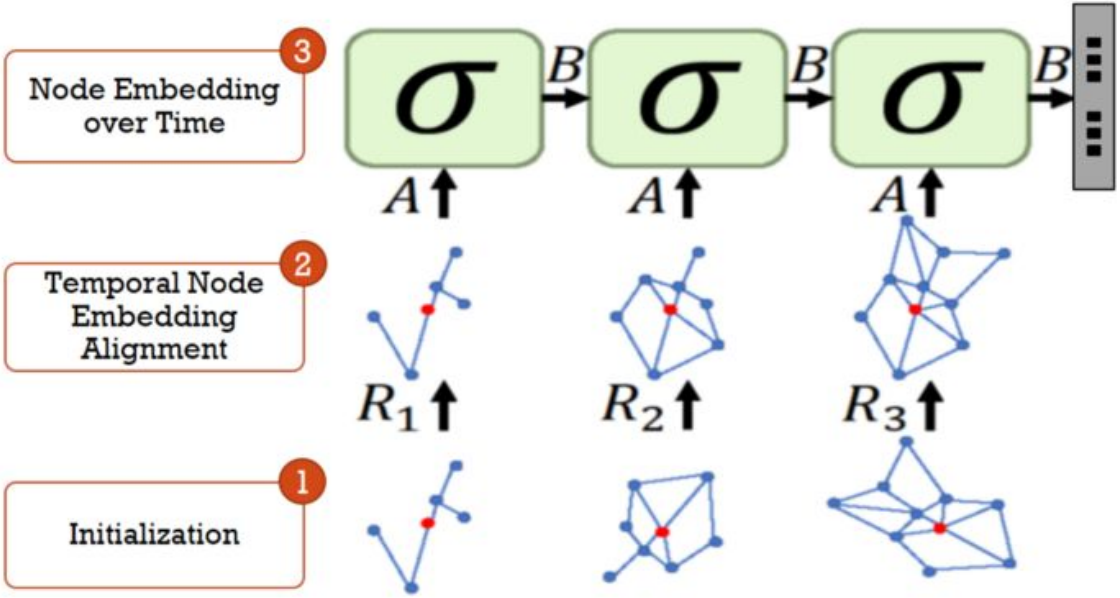







IHME
COVID-19 Projections
GHDX
VIZ HUB
Last updated June 8, 2020 (Pacific Time)
FAQ | Update notes | Article
United States of America
Trend
Compare
Map
Daily deaths
Today
2.4k
2.2k
2k
1.8k
1.6k
1.4k
1.2k
1k
800
600
400
200
0
Daily deaths
Mar 1
Apr 1
May 1
Jun 1
Jul 1
Aug 1
Date
August 04, 2020 * Projected
Daily deaths* 550 (264–1,203)
Daily deaths (reported)
Daily deaths (projected)
All deaths specific to COVID-19 patients.
Shaded area indicates 95% uncertainty interval.

Sequence tasks on graphs (many to many)



Mechanistic x Statistic models

Define your own rules or let the data do the talking?

Mechanistic x Statistic models approach a problem from different angles

- Do **you make the rules**?
- Or should you **let your juicy data learn the rules for you**?

Both types of models absolutely drive the world around us.

Mechanistic x Statistic models

Mechanistic models

- Mechanistic models are **based** on the fundamental **laws of natural sciences**. Physical and biochemical principles constitute the model equations. Few experimental data is needed to calibrate the model and determine unknown model parameters, such as adsorption coefficients, diffusivity or material properties.
- **differential equations** (DE), **numerical simulations**, **analytical solution**

Statistical models

- Statistical approaches utilize **statistics to predict trends and patterns**. All of these models learn from experience provided in the form of data. The more the experience, the better the model will be.
- **data-driven approaches, machine learning** (ML)

Example DE models

Navier-Stokes (meteorology)

- The model behind weather predictions. It is a chaotic model — meaning predictions can be wildly off when using just slightly incorrect inputs. That's why weather predictions are often wrong! Simulations are carried out with super-computers.

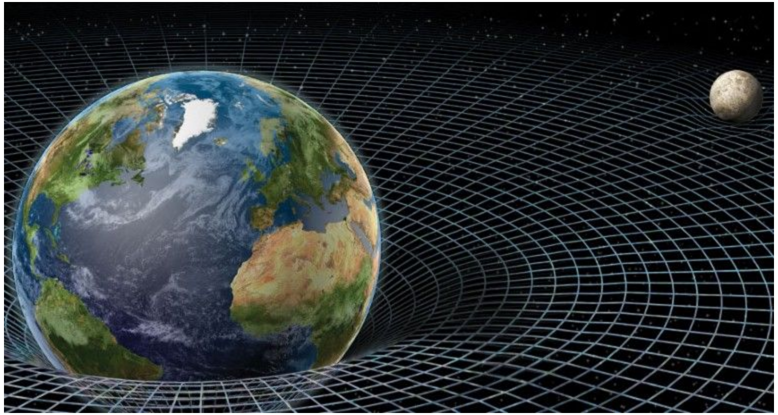
$$\overbrace{\rho}^{\substack{\textbf{DENSITY}\\ \text{of fluid}}} \overbrace{\left(\underbrace{\frac{\partial V}{\partial t}}_{\substack{\text{Change in}\\ \textbf{fluid velocity}\\ \text{over time}}} + \underbrace{V \cdot \nabla V}_{\substack{\text{The speed and}\\ \text{direction in which}\\ \text{fluid is moving}}}\right)}^{\substack{\textbf{ACCELERATION}\\ \text{of fluid}}} = \overbrace{\underbrace{\nabla P}_{\substack{\textbf{Internal}\\ \textbf{pressure}\\ \textbf{gradient}\\ \text{of fluid}}} + \underbrace{\rho g}_{\substack{\textbf{External}\\ \textbf{forces}\\ \text{on fluid}\\ \text{(gravity)}}} + \underbrace{\mu \nabla^2 V}_{\substack{\textbf{Internal stress}\\ \textbf{forces}\text{ acting}\\ \text{on fluid}\\ \text{(e.g. viscosity)}}}}^{\substack{\textbf{FORCES}\\ \text{acting on fluid}}}$$

Example DE models

Einstein Field Equations (physics)

- Describes governing laws behind gravity. Mathematical basis behind Einstein's General Theory of Relativity.

$$\overbrace{R_{uv} - \frac{1}{2} R\, g_{uv} + \Lambda\, g_{uv}}^{\substack{\textbf{CURVING} \\ \text{of spacetime} \\ \text{by matter \& energy}}} = \overbrace{\frac{8\pi G}{c^4} T_{uv}}^{\substack{\textbf{MOVEMENT} \\ \text{through spacetime} \\ \text{by matter \& energy}}}$$



Example DE models

Black-Scholes (finance)

- Prices financial derivatives at the stock market

price of **call** or **put option**

stock volatility

$$rV = \frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S}$$

change in call/option price over time

price of **stock**

Example DE models

SIR model (epidemiology)

- Basic compartmental model describing the spread of an infectious disease. Some recent applications to COVID-19: here, here, here & here.

Change in **susceptible** people over time: $$\frac{dS}{dt} = -\beta S I$$

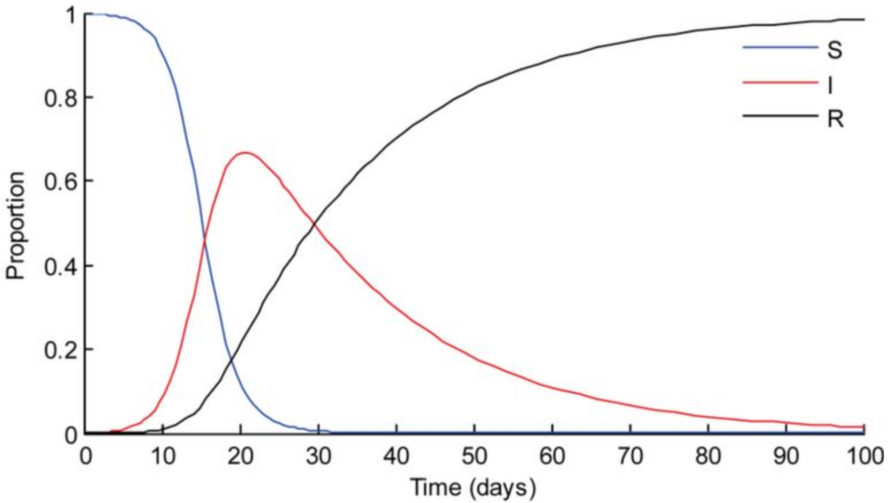
Change in **infected** people over time: $$\frac{dI}{dt} = \beta S I - \gamma I$$

Change in **recovered** people over time: $$\frac{dR}{dt} = \gamma I$$

infection from **interaction**

infectees **recover**

$$S \xrightarrow{\lambda} I \xrightarrow{\gamma} R$$



Murray-Gottman (psychology)

- Predicts the longevity of a romantic relationship… dang! Based on groundbreaking work by psychologist John Gottman, the model predicts that consistent positivity vibes is a strong predictor of marriage success.

wife's reaction

wife's mood when alone

wife's mood when with **husband**

husband's influence on **wife**

$$W_{t+1} = w + r_W W_t + I_{HW}(H_t)$$
$$H_{t+1} = h + r_H H_t + I_{WH}(W_t)$$

husband's reaction

husband's mood when alone

husband's mood when with **wife**

wife's influence on **husband**

Example DE models

Why are these four equations differential equations? Because they all contain derivatives (i.e. rate of change) of some unknown function. These unknown functions, e.g. S(t), I(t) and R(t) in the SIR model, are called solutions of the DE.

Note that Murray-Gottman's 'love model' is actually a **difference equation**, a type of sister model to **differential equations**. Difference equations output discrete sequences of numbers (e.g. census results every 5 years), while differential equations models continuous quantities — things which are happening all the time.

Mechanistic models

All five models we've seen so far — **differential and difference equations** — **are mechanistic models**, **where we choose the logic, rules, structure or mechanisms of the system ourselves**. (Of course, we don't always make the right choices. Trial and error plays a big part in mathematical modelling!)

- Navier-Stokes assume the atmosphere's just a moving fluid. Those equations come from fluid dynamics.
- General Relativity assume space-time warps under a special type of geometry. Here, clever chap Einstein came up with these profound ideas of the curvature of space and time through thought experiments and eminent mathematicians Emmy Noether and David Hilbert then helped him package those ideas into the Einstein Field Equations.
- The SIR model assume viruses spread through direct contact between infected and non-infected and that those who have the disease automatically recover at some fixed rate.

Mechanistic models

With mechanistic models, **observation and intuition inform the model's design**, while data is used to validate your assumptions later

All of this stands in contrast to empirical or data-driven models, where **we let the data do the talking from the very start**.

- This includes **machine learning**, **where algorithms learn the underlying logic or rules of the system by being fed enough quality examples**.

This is a sensible (**and often necessary**) **thing to do when** the mechanisms of a **system** are simply **too difficult to isolate and define by us** mere mortal human beings!

Mechanistic models make assumptions about the underlying mechanisms driving a system up-front. Mechanistic models are prolific in physics. In fact, the field of mathematical modelling started with the quest to unlock the fundamental dynamics behind planetary motion in the 17th century.

Statistical or data-driven modelling, typically machine learning, lets your juicy data learn the structure of the system for you. This is called fitting. ML is especially useful for complicated systems where we're really not sure how to separate signal from noise. No problem — just train a clever algorithm do the hard lifting for you. Ah, life's good. (Especially if you have a fast computer!)

Benefits of mechanistic x statistical models

An essential benefit of mechanistic vs. statistical models is that the **model parameters have an actual physical meaning**, which **facilitates the scientific interpretation of the results**.

Since natural laws are generally valid, mechanistic models are as well – even far beyond the calibration space. In practice, this means that you can easily change process parameters and the actual set-up freely

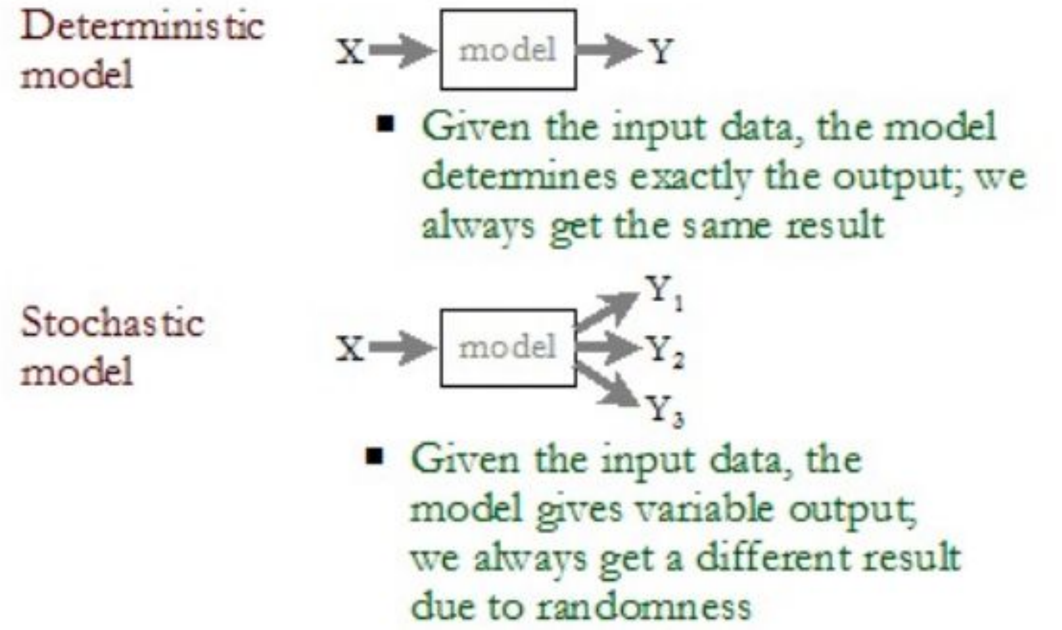
Opens a wide range of model applications using one and the same mechanistic model: from early-stage process development, process characterization and validation to process monitoring and control.

	MECHANISTIC MODELS	STATISTICAL MODELS
PRINCIPLE	Uses natural laws	Finds patterns in existing data
EQUATIONS	Complex equations from natural sciences	Simple equations, derived from statistics and regression analysis
DATABASE	Very few data needed (3-10 exp)	A lot of data needed (the more data, the better)
IMPLEMENTATION	Very high effort to program a simulation tool. Once a model is implemented and calibrated, very low cost of ownership	Low programming effort, low ownership cost
CALIBRATION EFFORT	Little effort to generate data for model calibration	Very high initial effort to generate data and initialize model
PROCESS FLEXIBILITY	yes	no
INTERPOLATION	yes	yes
EXTRAPOLATION	yes	no
GENERATES PROCESS UNDERSTANDING	yes	limited
QUALIFICATION DOWNSTREAM	Ideal for DSP, the same model is used throughout the process lifecycle	Sub-optimal for DSP, solves only one problem at a time
QUALIFICATION UPSTREAM	Very complex, only few examples of industrial application	Frequently used to guide process optimization and scale-up

Deterministic x stochastic

We can further classify mechanistic & empirical models into being either ‘deterministic’ (predictions are fixed) or ‘stochastic’ (predictions include randomness).

- **Deterministic models** ignore random variation and always predict the same outcome given the same starting conditions.
- **Stochastic models** take into account random variations, such as the heterogeneity of individual agents in a system. Examples include minor differences between people, animals, cells, etcetera.



Stochasticity

Stochasticity often introduces some realism into your model, but at a cost. In mathematical modelling, we always have to worry about model complexity.

- **Simple models** are **easy to analyse** but might **lack predictive power**.
- **Complicated models** are **realistic** but have fun trying to work out **what the hell's going on** behind the hood.
- So you end up with a **trade-off between simplicity and analytical tractability**. The delicate act of balancing the two is a dark art of sorts. Like statistician George Box said:

  All models are wrong, but some are useful.

-

Stochasticity

In machine learning and statistics, model complexity is known as the bias-variance tradeoff.

- High-bias models are too simple, resulting in underfitting
- High-variance models memorise the noise (i.e. distractions) instead of the signal (i.e. real structure of the system), causing overfitting.

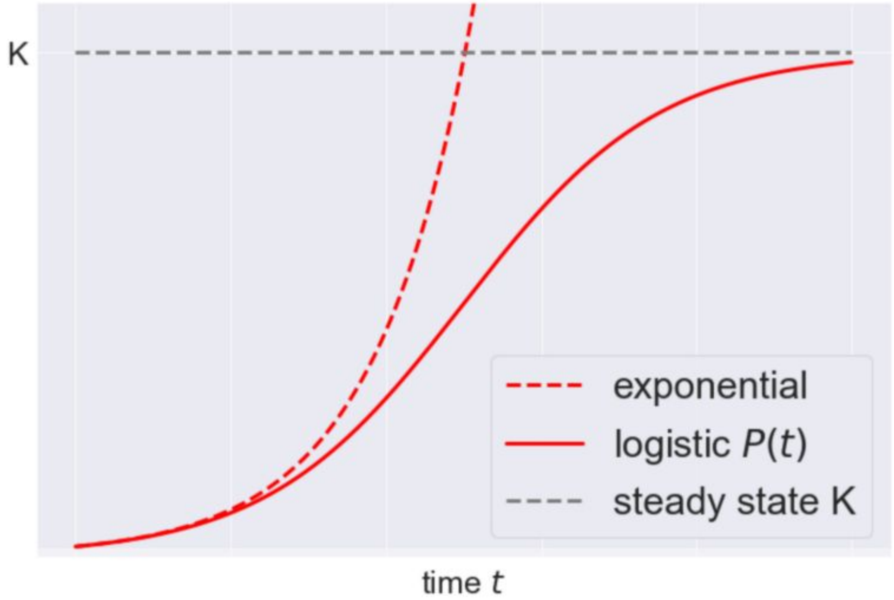
The dark art of balancing this model complexity is often what separates a good from a not-so-good data scientist

Comparative example - DE vs ML

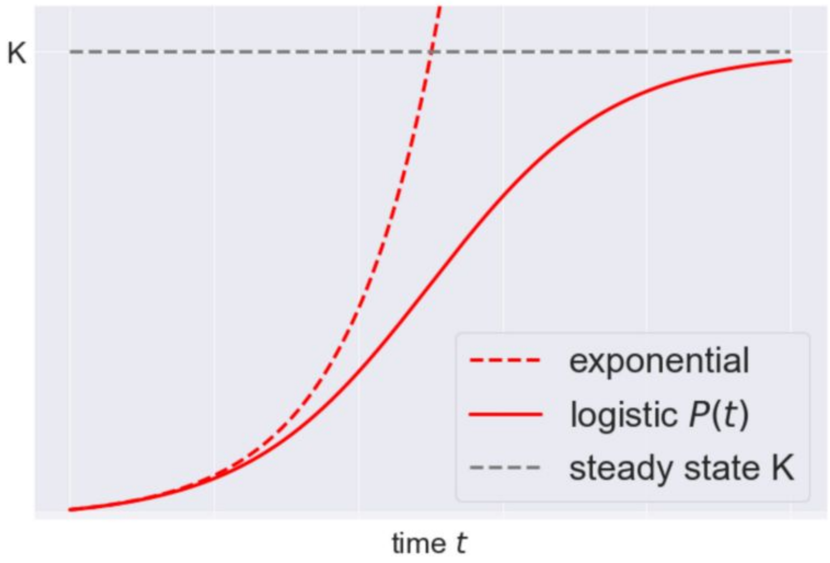
Consider the **logistic differential equation**, which **models resource-limited growth in many different disciplines** — agriculture, biology, economics, ecology, epidemiology, and so on. It's a real swiss-army knife model.

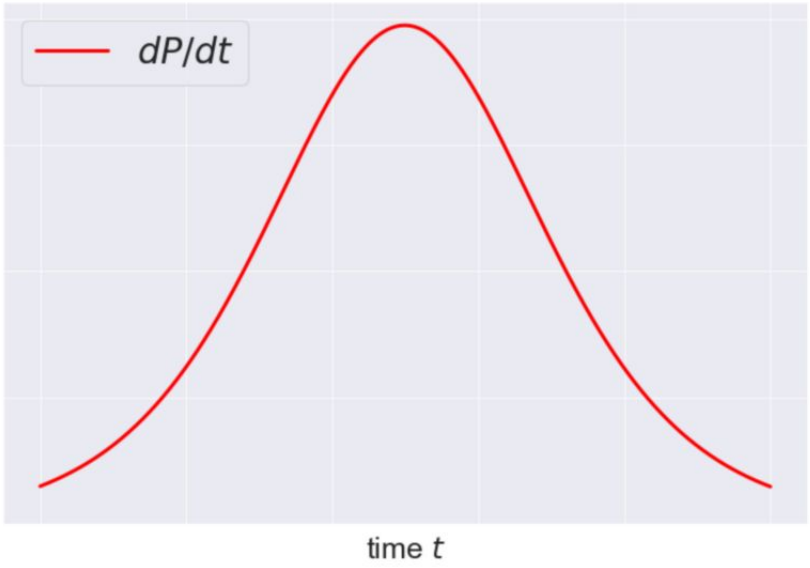
Change in **logistically-growing** quantity over time:

$$\frac{dP}{dt} = \underbrace{r\,P}_{\substack{\textbf{exponential} \\ \text{growth} \\ \text{term}}} \underbrace{\left(1 - \frac{P}{K}\right)}_{\substack{\text{forces long-term} \\ \text{behaviour to} \\ \textbf{carrying capacity } \boldsymbol{K}}}$$



logistic differential equation





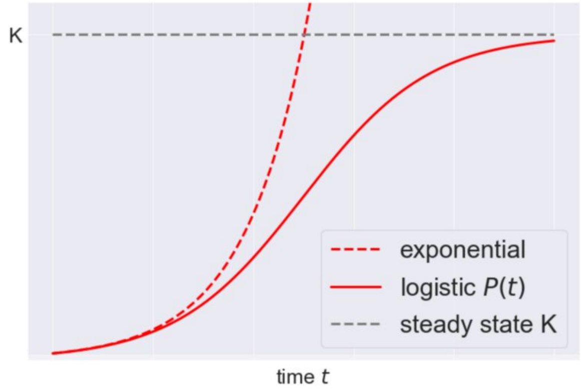
Logistic model example: Hubbert’s Peak Oil model

The time was 1956. Marion Hubbert, a Shell geophysicist, made a **predictive mathematical model for the amount of oil production in Texas**.

- Let P represent the amount of oil produced in Texas.
- If the right-hand side was just rP, then oil production would grow exponentially.
- However Hubbert knew there was only K = 200 gigabarrels of oil in total. As time goes on, it becomes harder to extract oil, which then decreases the production rate dP/dt.
- The inclusion of the (1 — P/K) term accounts for this resource-limited observation. **Notice that we have inferred the mechanisms of oil extraction before any real consideration of real data**.

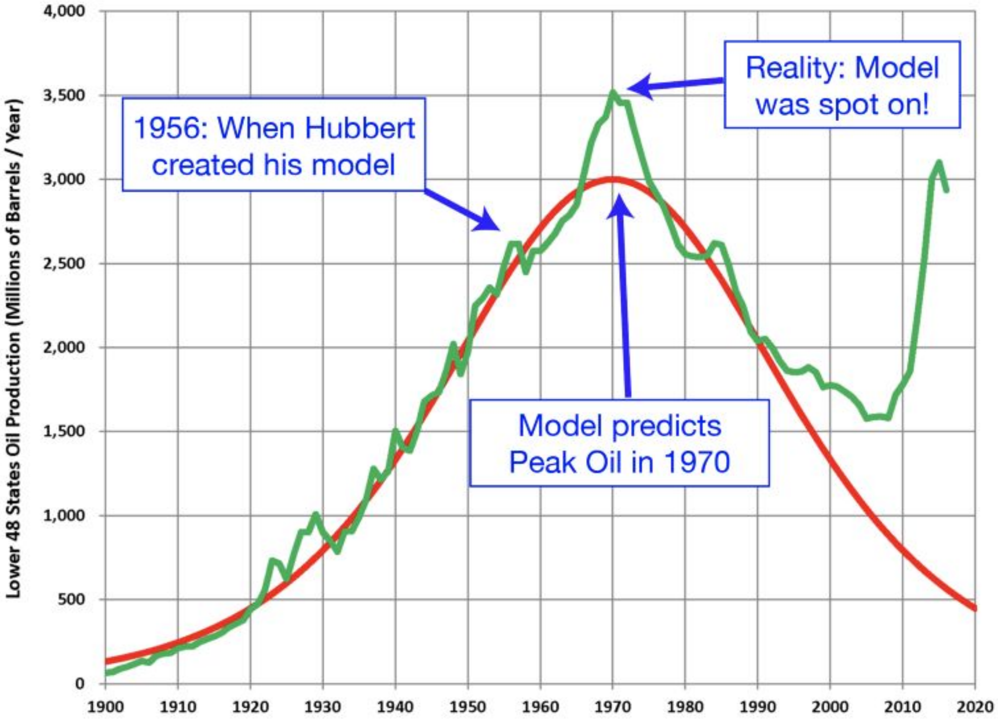
Change in **logistically-growing** quantity over time:

$$\frac{dP}{dt} = \underbrace{r\,P}_{\substack{\textbf{exponential}\\ \text{growth}\\ \text{term}}} \underbrace{\left(1 - \frac{P}{K}\right)}_{\substack{\text{forces long-term}\\ \text{behaviour to}\\ \textbf{carrying capacity } \boldsymbol{K}}}$$



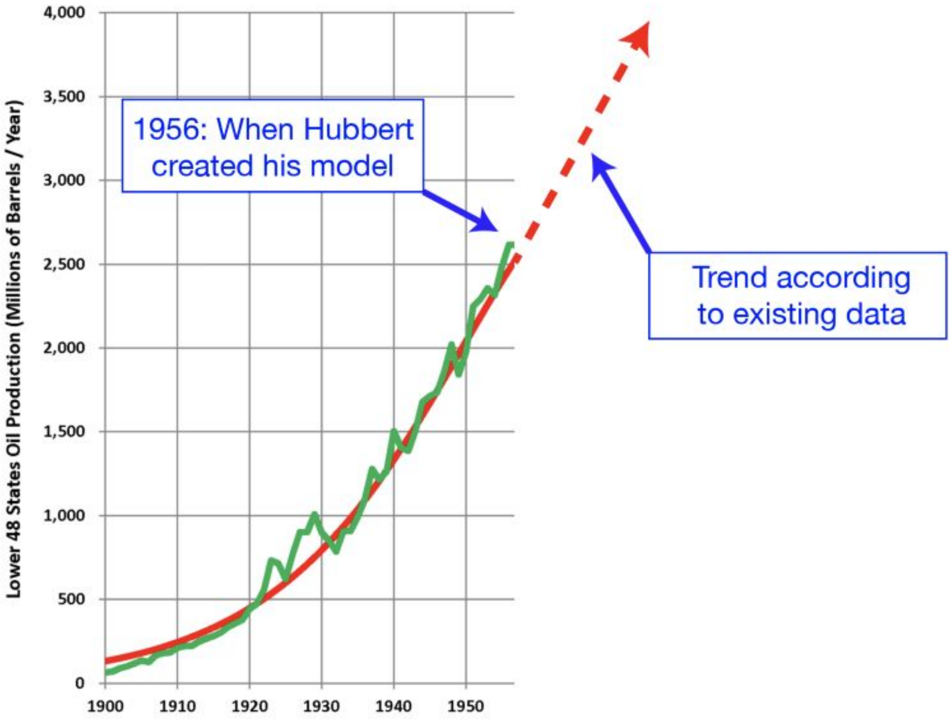
Logistic model example: Hubbert’s Peak Oil model

- The parameter r = 0.079, representing the production rate, was extrapolated from 50 years worth of data.
- The parameter K = 200, represents the total amount of oil in the ground. It is a steady state of the system.
- Hubbert’s model **predicted that Peak Oil** (the time of maximum oil production) **would occur 14 years later in 1970**. It turns out his model also nailed the decline
- The spike after 2000 was due to Alaskan contributions that weren’t part of Hubbert’s original modelling scope. Hubbert would have been pleased.



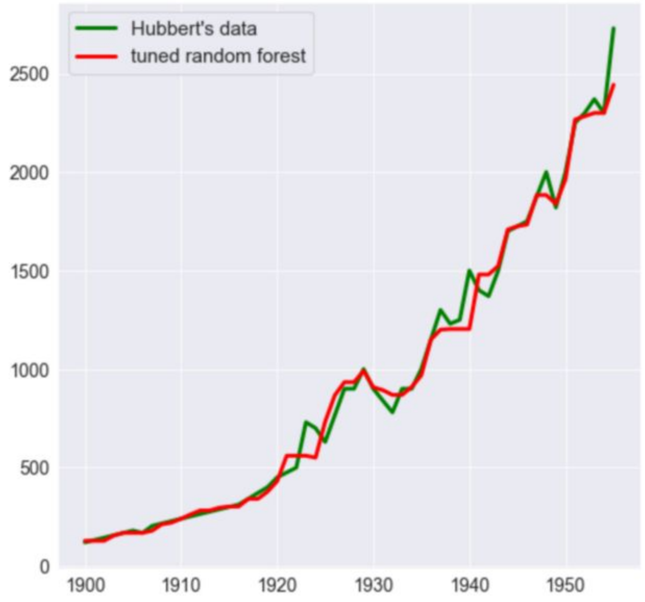
Logistic model example: Hubbert's Peak Oil model

- A machine learning model would have a **very difficult time learning the underlying mechanisms captured by the logic embedded into differential equation**.
- Essentially, any algorithm would need to **predict a maximum to occur with just the data (in green) existing only until 1956**:



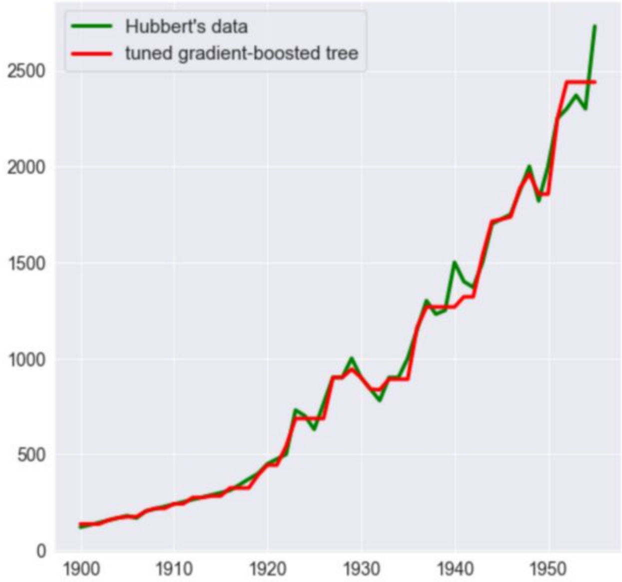
Logistic model example: Hubbert's Peak Oil model

- Trying some polynomial regressions, random forests
- Note that only the polynomial regression will extrapolate beyond the original data range!

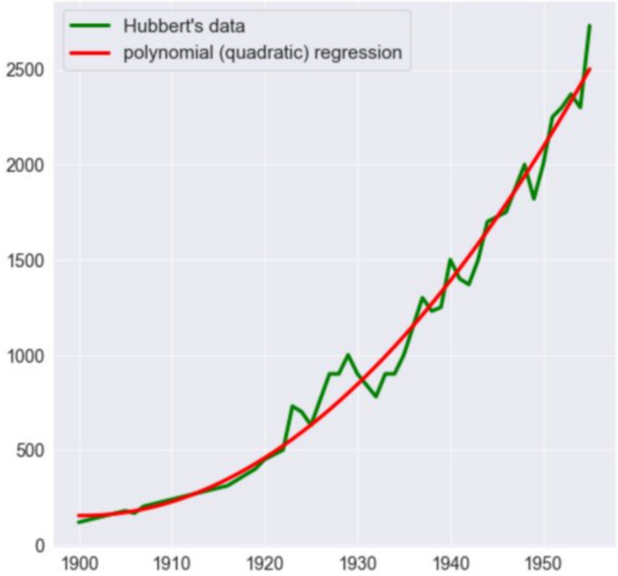



```
Performance on TRAINING set
----------------------------------
MAE:  36.89
MSE:  67.7
R2:  0.99

Performance on TEST set
----------------------------------
MAE:  68.97
MSE:  101.47
R2:  0.98
```

Logistic model example: Hubbert's Peak Oil model




```
Performance on TRAINING set
---------------------------------
MAE:  28.81
MSE:  57.97
R2:  0.99

Performance on TEST set
---------------------------------
MAE:  68.28
MSE:  94.65
R2:  0.98
```




```
Performance on TRAINING set
---------------------------------
MAE:  60.09
MSE:  86.5
R2:  0.99

Performance on TEST set
---------------------------------
MAE:  67.0
MSE:  84.45
R2:  0.98
```

Logistic model example: Hubbert's Peak Oil model

- The polynomial regression captures the signal well, but it would be impossible for this quadratic function (i.e. parabola) to concave back after hitting Peak Oil in 1970. The red curve would simply climb higher and higher, corresponding to oil production approaching infinity!
- **Hubbert's mechanistic take won this modelling challenge.**

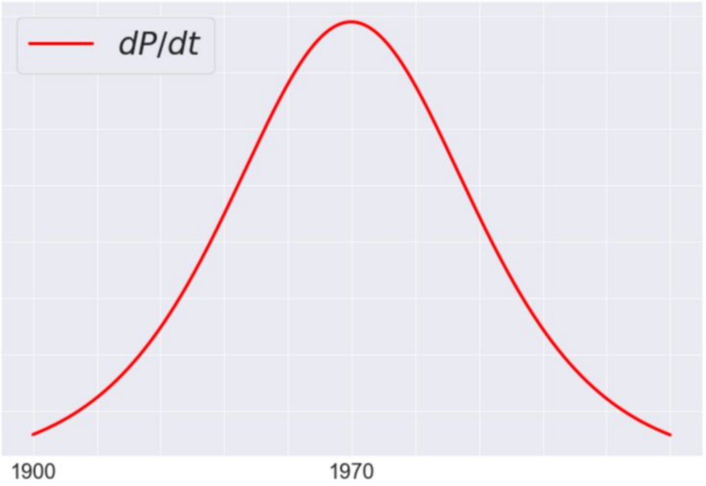
- **Machine learning shines when the rules and governing dynamics of a system are too difficult to capture and define by humans**.
- A fine approach is to then simply let a machine learn these rules and signals through the consumption of quality examples.
- We are training the machine with data, t**he better the data, the better the results**.

Solving the logistic differential equ

Approach 1: numerical simulations

One can program the differential equation into Python or Matlab, use a numerical solver to obtain P(t) before plotting dP/dt as a function of t. Here, I used Python.

```
import numpy as np
from scipy.integrate import odeint
from sympy import *
import matplotlib.pyplot as plt
import seaborn as sb
sb.set_style('darkgrid')

# model
def model(y,t):
    k = 0.3
    dydt = r * y * (1 - y/K)
    return dydt

# initial condition
y0 = 3.8

# parameters
r = 0.079
K = 200

# time points
t = np.linspace(0,100,10000)

# ODE solver
y = odeint(model, y0, t)

# plot results
plt.figure(figsize=(12,8))
plt.plot(t,model(y,t), linewidth=3, color='red') # sketch derivative (t, dydt)
plt.legend(["$dP/dt$"], loc=2, prop={'size': 30})
plt.show()
```



Solving the logistic differential equation

Approach 2: obtain the analytic solution

This system can be solved analytically using separation of variables.

- Note that the vast majority of DE's cannot be solved analytically!
- Mathematicians out there spend their entire working lives searching for analytic solutions. Take Roy Kerr, the New Zealand scientist who found an illusive solution of Einstein's equations leading to the discovery of black holes.

Thankfully, the logistic DE — along with many others taught in an intro-level DE course — can be explicitly solved

Solving the logistic differential equation

Approach 2: obtain the analytic solution

- We first move all the P terms to the left-hand side and the t terms to the right.

$$\int \frac{K}{P(K-P)}\,dP = \int r\,dt$$

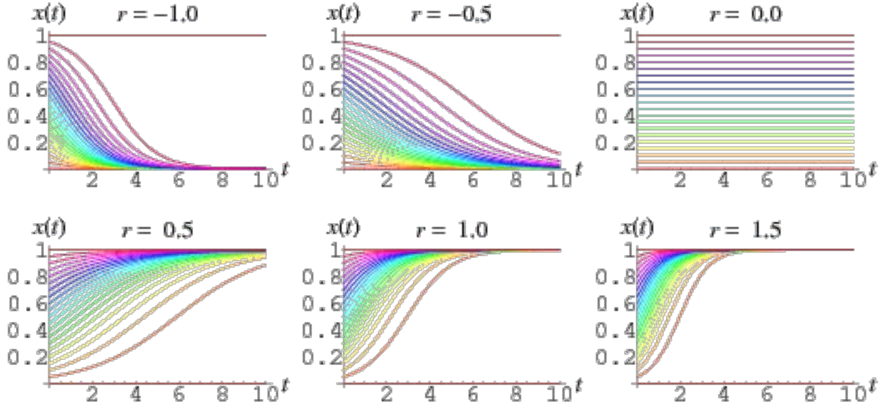
- Integrating both sides then gives us the general solution, which is an infinite set of functions that satisfies the DE.

$$\ln(P) - \ln(K-P) = rt + C$$

- There are always an infinite number of solutions for a differential equation, graphically given by a family of curves.

Rearranging for P, we get:

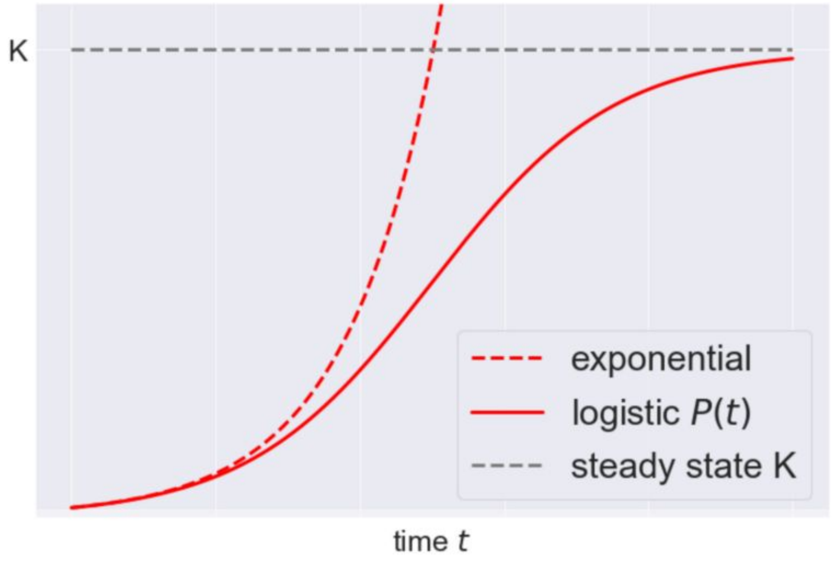
$$P(t) = \frac{K}{1 + Be^{-rt}}$$



Solving the logistic differential equation

These two formulas plot to give the logistic and Gaussian-like curves shown earlier.

$$P(t) = \frac{K}{1 + Be^{-rt}}$$



$$\frac{dP}{dt} = rK\frac{Be^{-rt}}{(1 + Be^{-rt})^2}$$