Machine Learning Techniques in Chemistry

Prof. Luciano T. Costa | ltcosta@id.uff.br
Prof. Eduardo Krempser | eduardo.krempser@fiocruz.br

molmod
PIBSS
Plataforma Institucional
Biodiversidade e Saúde Silvestre
Ministério da Saúde
FIOCRUZ
Fundação Oswaldo Cruz

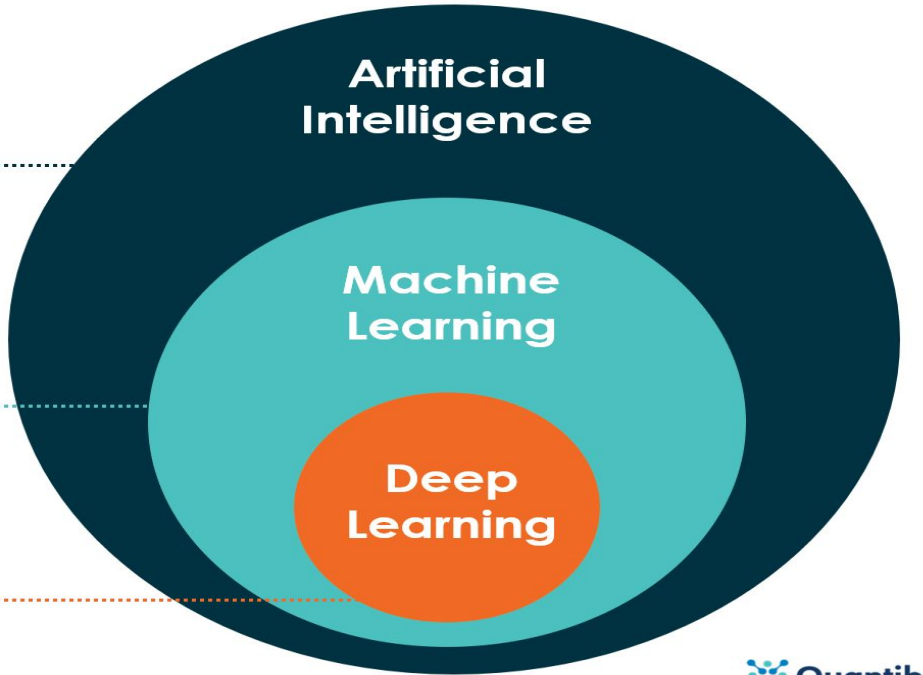
What is Artificial Intelligence?

"AI is an attempt to reproduce intelligence reasoning using machines."

ARTIFICIAL INTELLIGENCE (AI)
Programming systems to perform tasks which usually require human intelligence.

MACHINE LEARNING (ML)
Training algorithms to solve tasks by pattern recognition instead of specifically programming them how to solve the task.

DEEP LEARNING (DL)
Training algorithms by using deep neural networks with multiple layers.



Machine Learning

“The term “machine learning” is sometimes thrown around as if it is some kind of magic pill: apply machine learning to your data, and all your problems will be solved! As you might expect, the reality is rarely this simple.”

Python Data Science Handbook

What is Artificial Intelligence?

"AI is an attempt to reproduce intelligence reasoning using machines."

Python modules that can help to build machine learning codes:

```
NUMPY
SCIPY
SCIKIT-LEARN
```

https://scikit-learn.org/stable/install.html

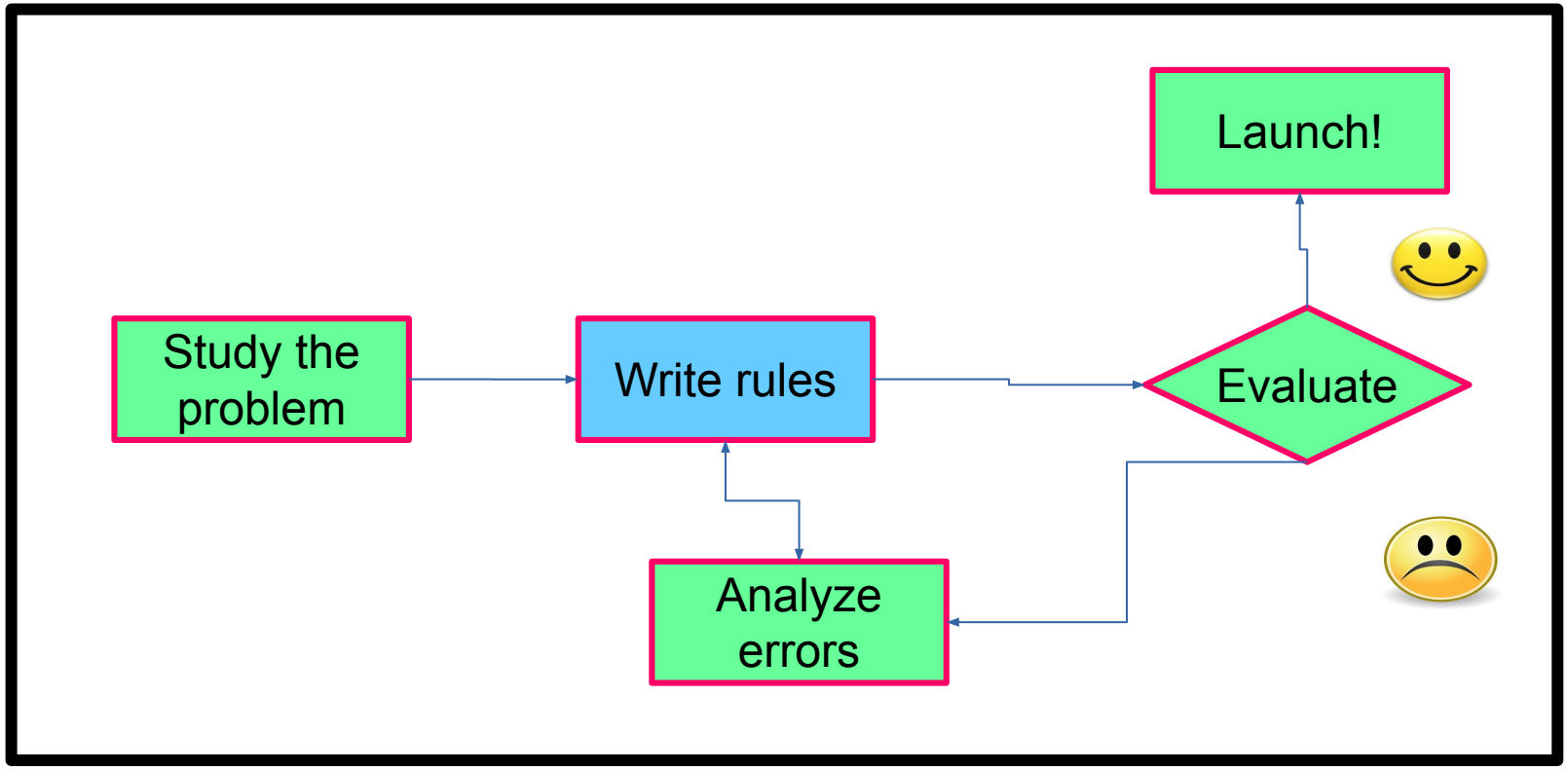
```
> pip install numpy scipy scikit-learn
```

- The process of training and prediction involves the use of specialized algorithms. We feed the training data to an algorithm, and the algorithm uses this training data to give predictions on a new test data.

- Arthur Samuel way back in 1959: "[**Machine Learning is the**] field of study that gives computers the ability to learn without being explicitly programmed."

- "A computer program is said **to learn from experience** E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E." (Tom Mitchell, 1997, Carnegie Mellon University)

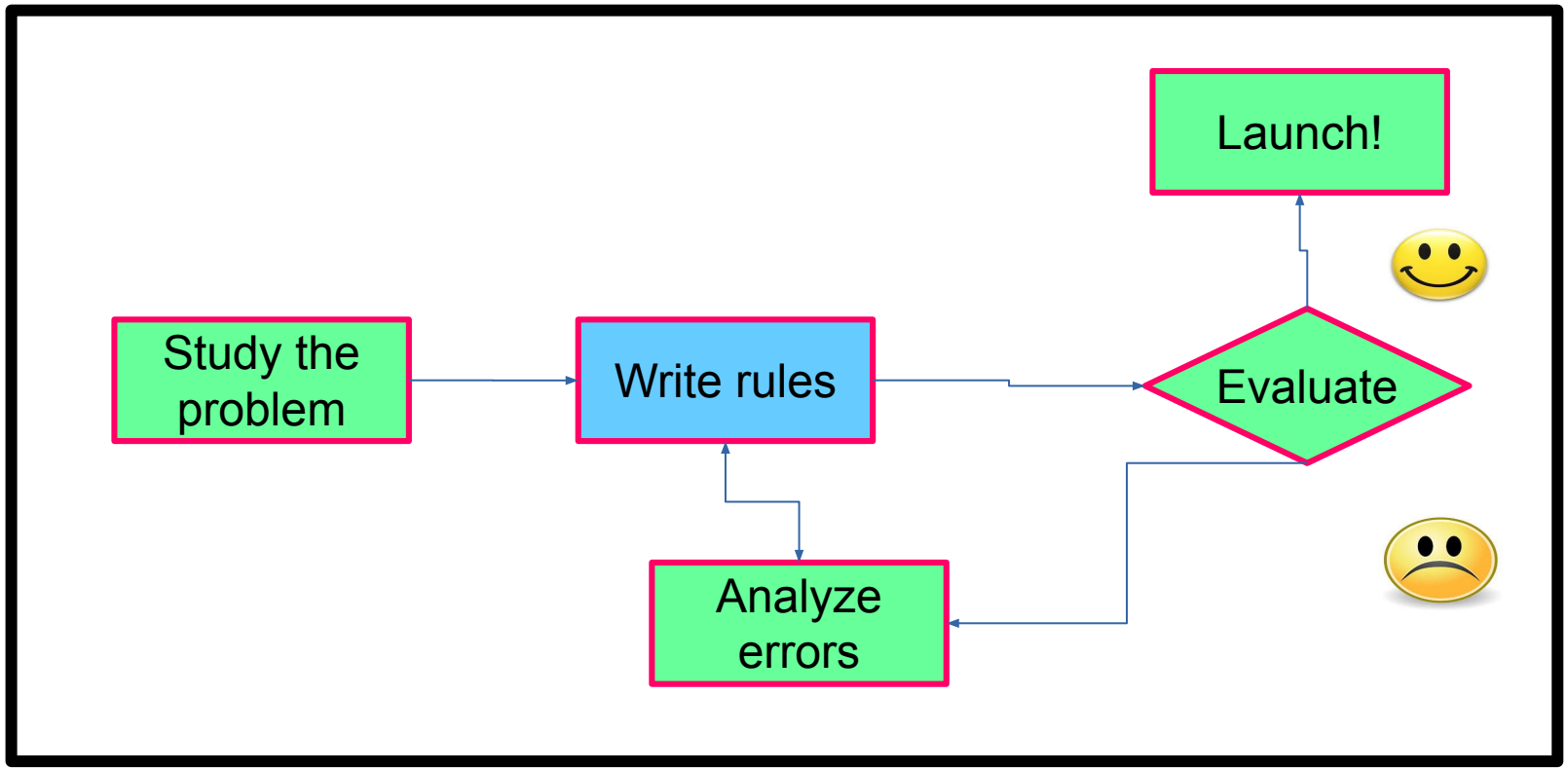
Traditional Approach



"Hands on Machine Learning", Aurélien Géron, O'Reilly

Machine Learning: How does it work?

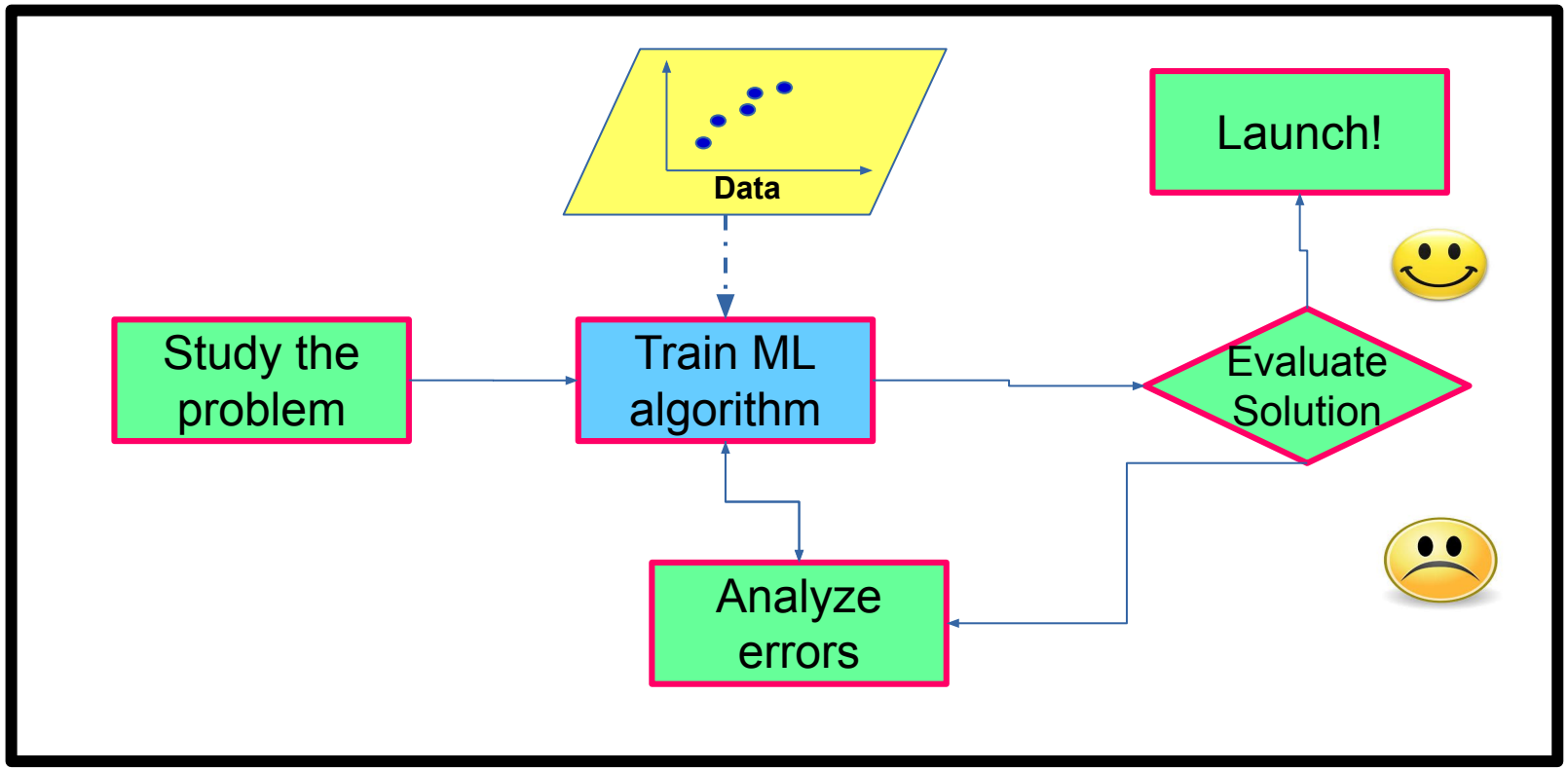
"Machine Learning is the science (and art) of programming computers so they can learn from data."



"Hands on Machine Learning", Aurélien Géron, O'Reilly

Machine Learning: How does it work?

"Machine Learning is the science (and art) of programming computers so they can learn from data."



"Hands on Machine Learning", Aurélien Géron, O'Reilly

Machine Learning: How does it work?

To summarize, **Machine Learning** is great for:

• Problems for which existing solutions require a lot of hand-tuning or long lists of rules: one Machine Learning algorithm can often **simplify code and perform better**.

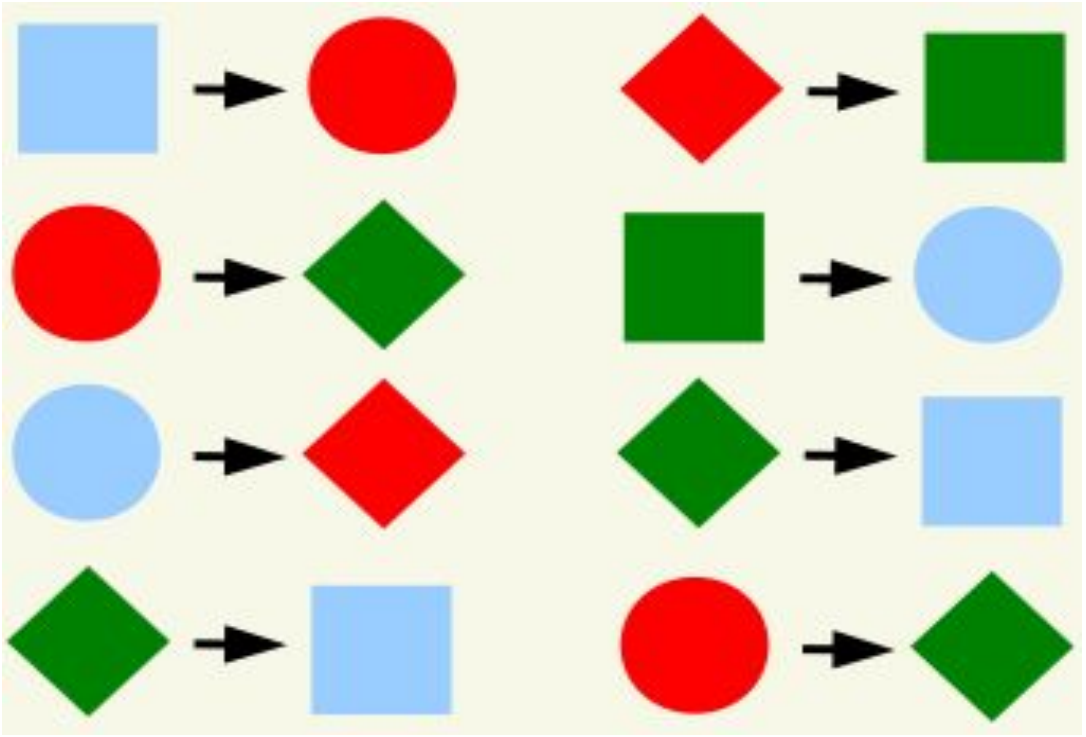
• Complex problems for which there is no good solution at all using a traditional approach: the best **Machine Learning techniques can find a solution**.

• Fluctuating environments: a Machine Learning system **can adapt** to new data.

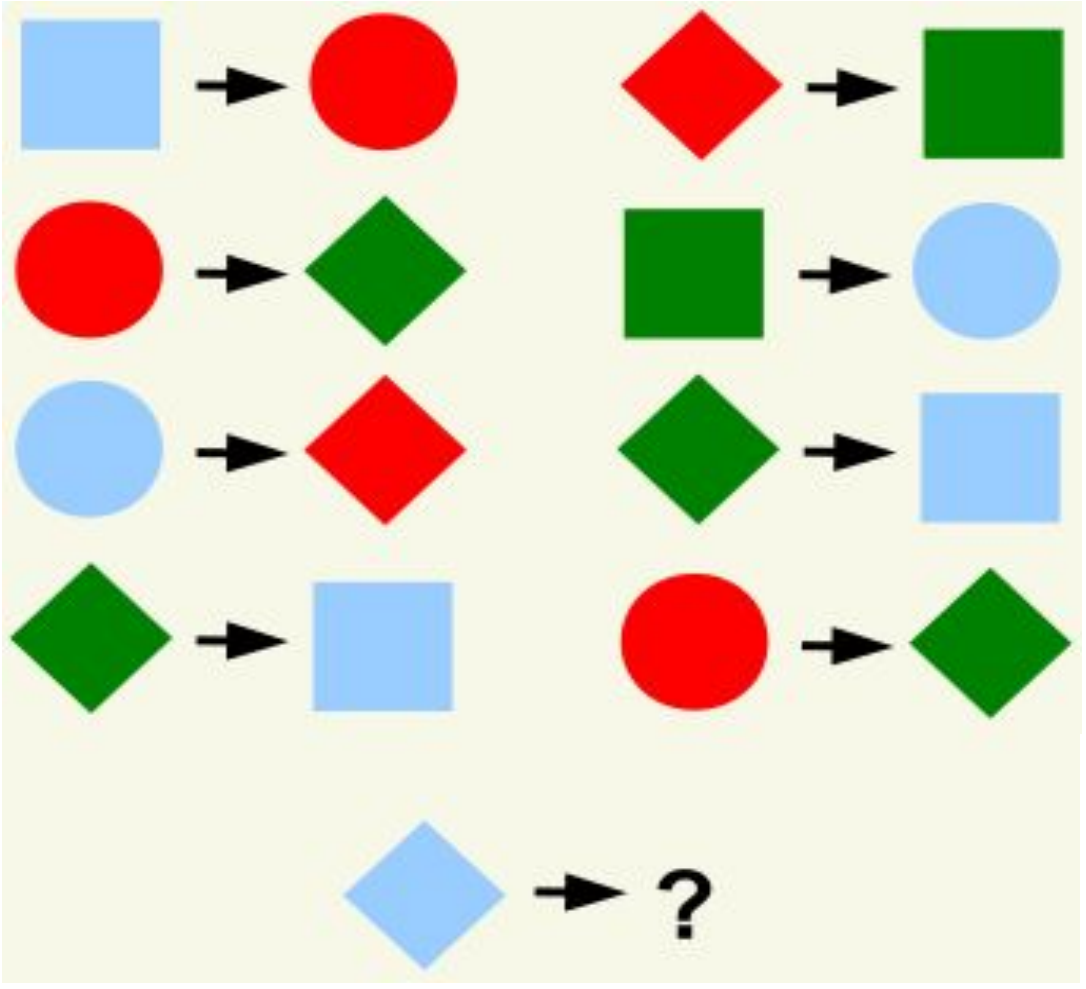
• Getting insights about complex problems and large amounts of data.

"Hands on Machine Learning", Aurélien Géron, O'Reilly

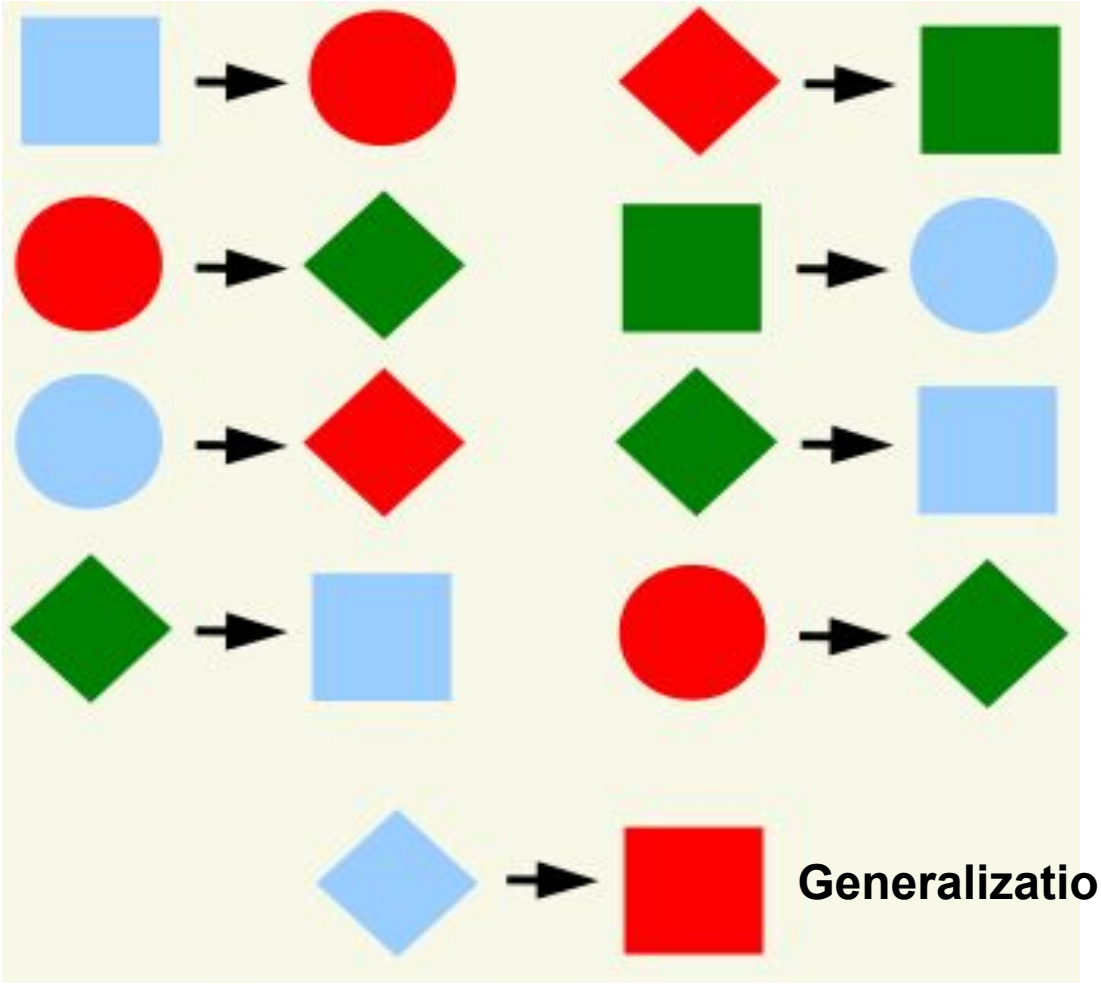
Machine Learning: How does it work?



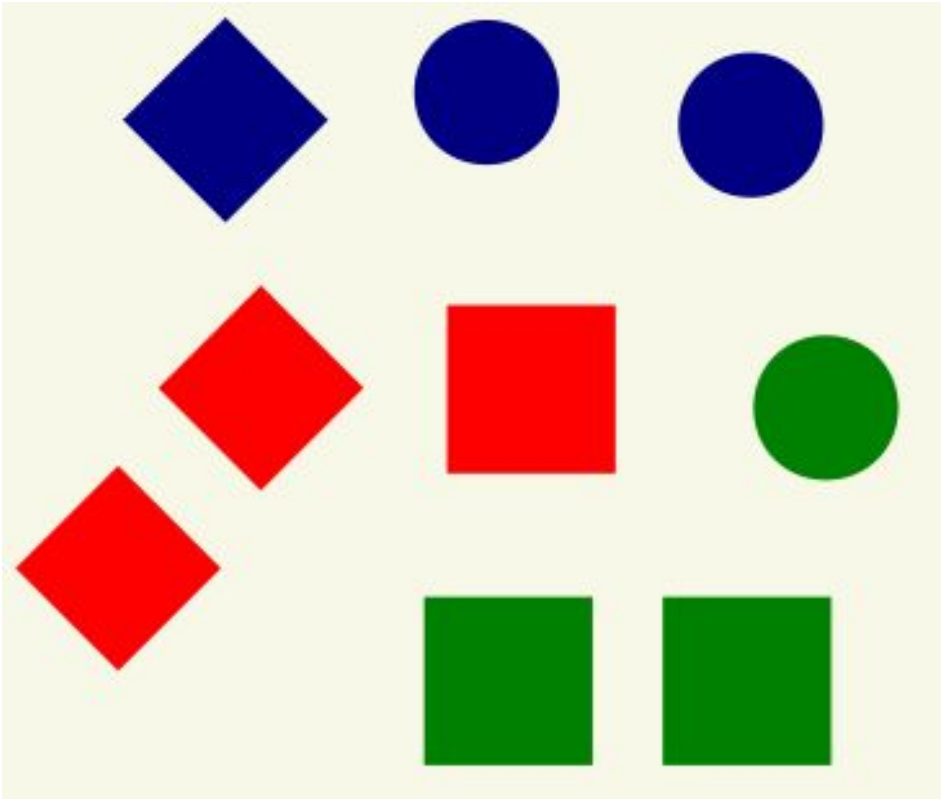
Machine Learning: How does it work?



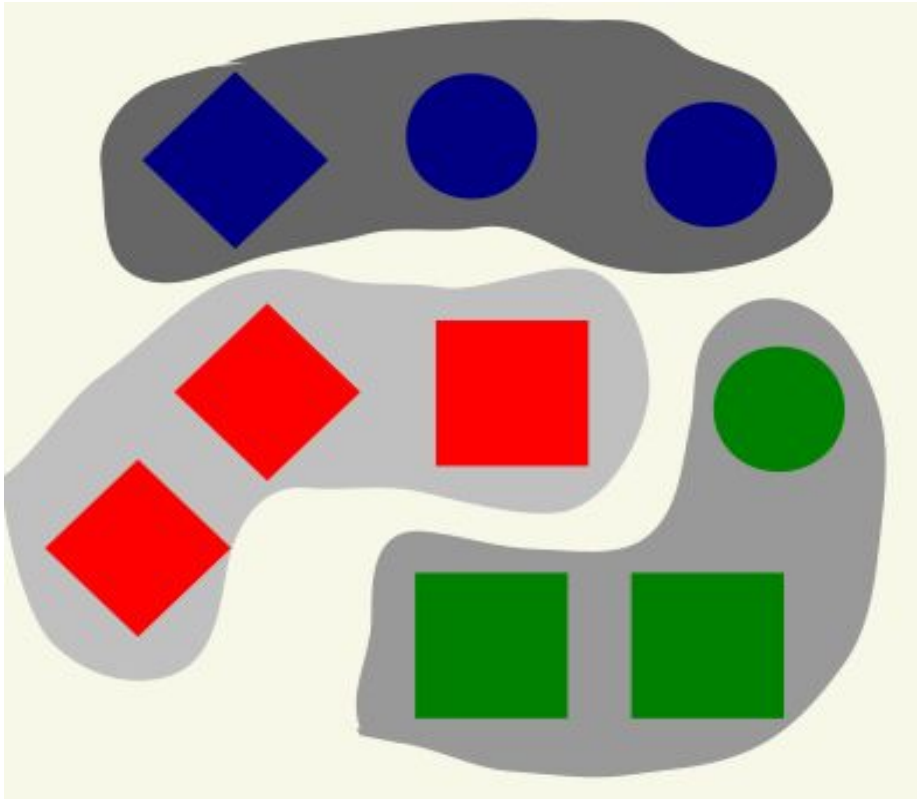
Machine Learning: How does it work?



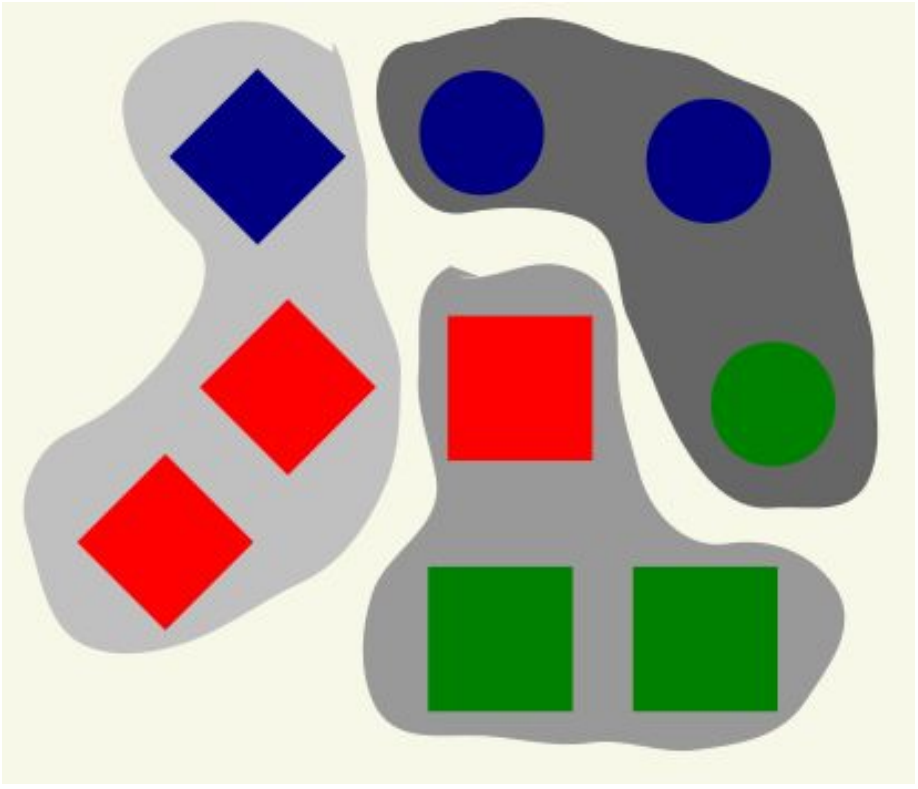
Machine Learning: How does it work?



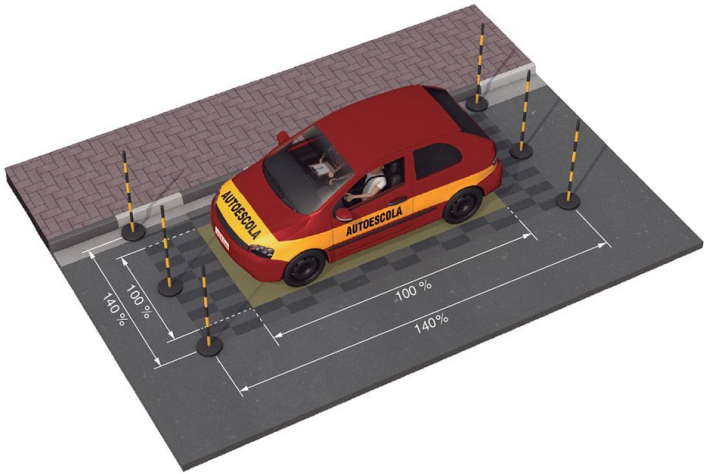
Machine Learning: How does it work?



Machine Learning: How does it work?

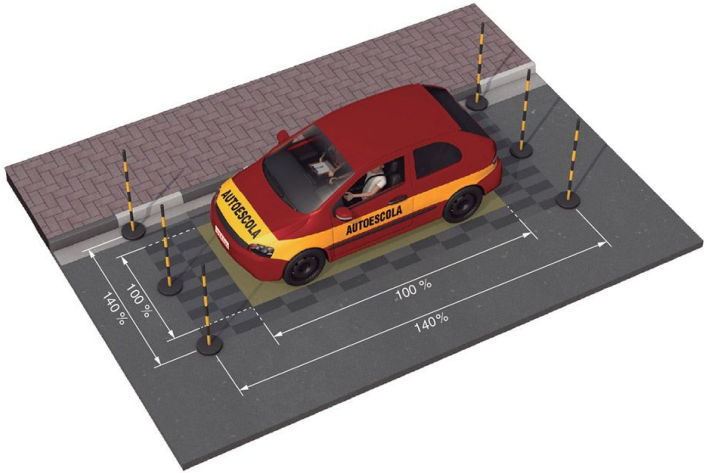


Machine Learning: How does it work?



Imagens:
https://www.autoescolaonline.net/como-fazer-uma-baliza-perfeita-em-10-passos
https://unsplash.com/photos/q88ZVP2f2fg

Machine Learning: How does it work?

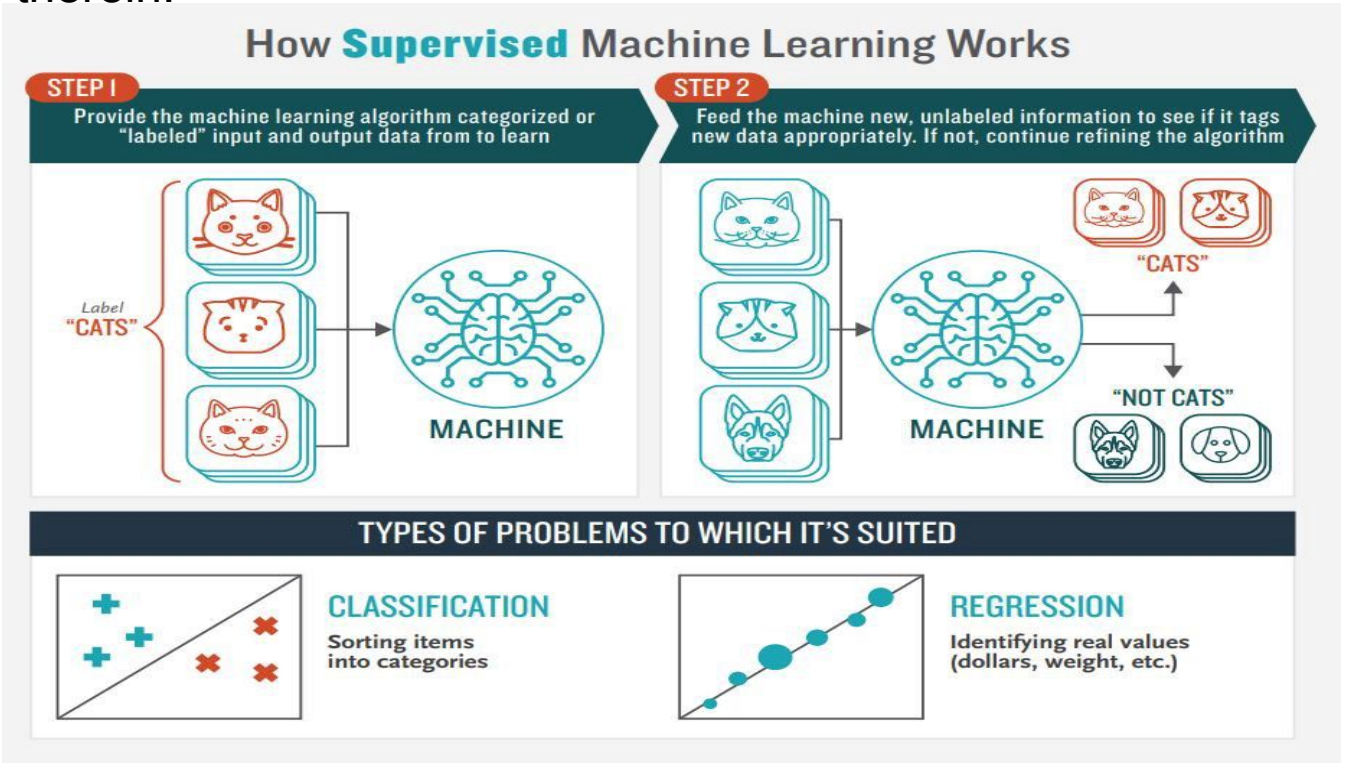


Imagens:
https://www.autoescolaonline.net/como-fazer-uma-baliza-perfeita-em-10-passos
https://unsplash.com/photos/q88ZVP2f2fg

Types of ML tasks

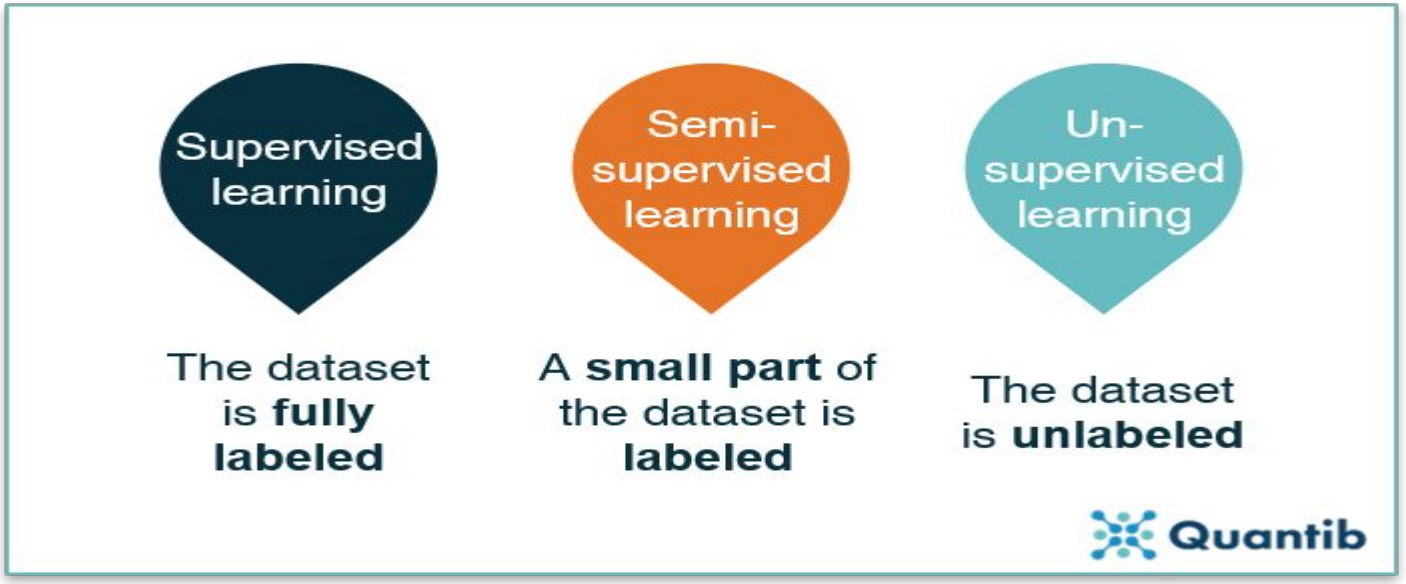
Supervised ML: The program is “trained” on a pre-defined set of “training examples”, which then facilitate its ability to reach an accurate conclusion when given new data.

Unsupervised ML: The program is given a bunch of data and must find patterns and relationships therein.



Types of ML tasks

Instead, a **semi-supervised** strategy can be used with a subset of manually labeled images (e.g. with a segmentation of white matter hyperintensities), which can be combined with a bigger unlabeled dataset



Supervised ML - Example problem

Location	Bedrooms	Car Spaces	...	Price (R$/m^2)
Copacabana	2	Não	...	41,96
Copacabana	1	Sim	...	49,15
Centro	2	Não	...	28,05
Barra da Tijuca	3	Sim	...	30,86
...	...	...	...	...

Supervised ML - Example problem

Location	Bedrooms	Car Spaces	...	Price (R$/m^2)
Copacabana	2	Não	...	41,96
Copacabana	1	Sim	...	49,15
Centro	2	Não	...	28,05
Barra da Tijuca	3	Sim	...	30,86
...	...	...	...	...

Location	Bedrooms	Car Spaces	...	Price (R$/m^2)
Centro	1	Sim	...	**?**

Supervised ML - Example problem

Location	Bedrooms	Car Spaces	...	Price (R$/m^2)	Rent?
Copacabana	2	Não	...	41,96	Yes
Copacabana	1	Sim	...	49,15	No
Centro	2	Não	...	28,05	No
Barra da Tijuca	3	Sim	...	30,86	Yes
...	...	...	...	...	...

Location	Bedrooms	Car Spaces	...	Price (R$/m^2)
Centro	1	Sim	...	**?**

Supervised ML - Example problem

Regression

Classification

Location	Bedrooms	Car Spaces	...	Price (R$/m^2)	Rent?
Copacabana	2	Não	...	41,96	Yes
Copacabana	1	Sim	...	49,15	No
Centro	2	Não	...	28,05	No
Barra da Tijuca	3	Sim	...	30,86	Yes
...	...	...	...	...	...

Location	Bedrooms	Car Spaces	...	Price (R$/m^2)
Centro	1	Sim	...	**?**

Supervised ML - Example problem

Location	Bedrooms	Car Spaces	...	Price (R$/m^2)
Copacabana	2	Não	...	41,96
Copacabana	1	Sim	...	49,15
Centro	2	Não	...	28,05
Barra da Tijuca	3	Sim	...	30,86
...	...	...	...	...

X **y**

Supervised ML - Example problem

Title	ALogP	ALogp2	AMR	BCUTp-1	BCUTp-1h	fragC	apol	naAromAt	nAtom
Acebutolol	-2.1004	4.41168	67.0727	4.75506	9.76616	2152.06	55.7582	6	52
Amoxicillin	-1.874	3.51188	63.5964	4.78119	12.1043	1516.09	51.0391	6	44
Bendroflumethlazide	-0.8075	0.65206	39.6471	4.24449	12.8541	1147.12	49.7141	12	41
Benzocaine	-0.4883	0.23844	20.5043	4.54796	8.52863	397.03	25.8787	6	23
Benzthlazide	-1.3398	1.79506	50.9651	6.69282	12.9721	1114.11	53.1231	10	40
Clozapine	0.0092	8.46E-05	45.0577	5.92358	11.5903	1519.05	50.9291	12	42
Dibucaine	-0.5602	0.31382	57.7981	5.09923	10.5046	2425.05	59.441	10	54
Diethylstibestrol	-0.2302	0.05299	33.7086	4.85094	11.5338	1301.02	46.6199	12	40
Diflunisal	0.2593	0.06724	11.5289	4.03982	9.84339	423.05	31.7343	12	26
Dipyridamole	-3.3772	11.4055	95.8664	4.30728	11.5724	4981.12	80.9197	10	76
Folic_Acid	-3.3182	11.0105	52.3395	3.83215	9.33155	1817.13	58.6211	12	51
Furosemide	-0.9949	0.98983	33.6078	4.6696	12.2524	669.09	39.7447	11	32
Hydrochlorothiazide	-1.5436	2.3827	35.4327	7.26833	12.8018	404.1	32.1423	6	25
Imipramine	0.091	0.00828	39.2562	5.70631	11.5894	1789.02	51.643	12	45
Indomethacin	0.2922	0.08538	37.8739	4.12177	11.086	1249.06	50.5967	12	41
Ketoprofen	-0.2088	0.0436	21.9194	4.29006	9.91827	814.03	39.9011	12	33
Lidocaine	1.166	1.35956	45.8867	5.09304	9.98654	1249.03	42.3114	6	39
Meclofenamic_acid	1.4704	2.16208	27.2526	4.4468	11.3886	619.05	39.0387	12	30

Machine learning methods in chemoinformatics. Mitchell, J. B. O.; 2014

Machine Learning: How does it work?

Some **regression algorithms** can be used for classification as well, and vice versa. For example, *Logistic Regression* is commonly used for classification, as it can output a value that corresponds to the probability of belonging to a given class.

Examples of SUPERVISED LEARNING ALGORITHMS

- k-Nearest Neighbors
- Linear Regression
- Logistic Regression
- Support Vector Machines (SVMs)
- Decision Trees and Random Forests
- Neural networks*

"Hands on Machine Learning", Aurélien Géron, O'Reilly

Machine Learning: How does it work?

Examples of UNSUPERVISED LEARNING ALGORITHMS

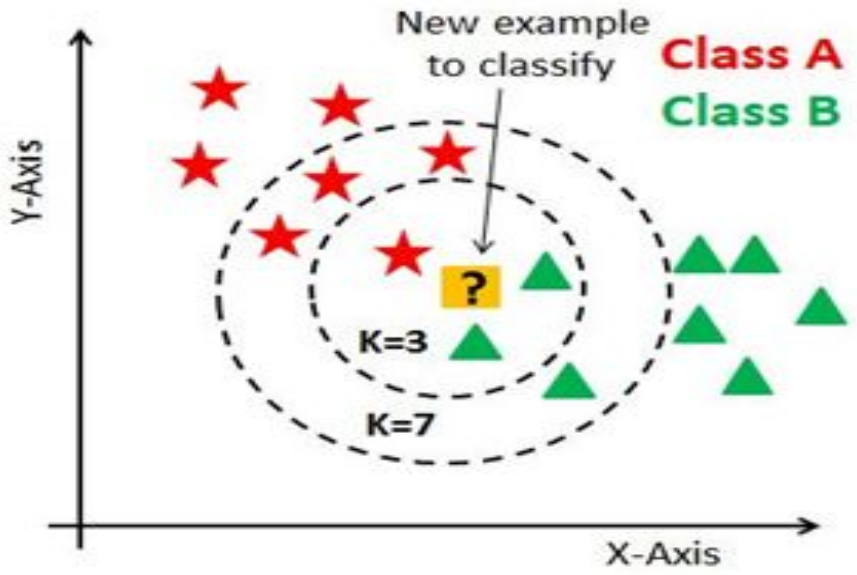
- Clustering
  - k-Means
  - Hierarchical Cluster Analysis (HCA)
  - Expectation Maximization
- Visualization and dimensionality reduction
  - Principal Component Analysis (PCA)
  - Kernel PCA
  - Locally-Linear Embedding (LLE)
  - t-distributed Stochastic Neighbor Embedding (t-SNE)

“Hands on Machine Learning”, Aurélien Géron, O’Reilly

Machine Learning: How does it work?

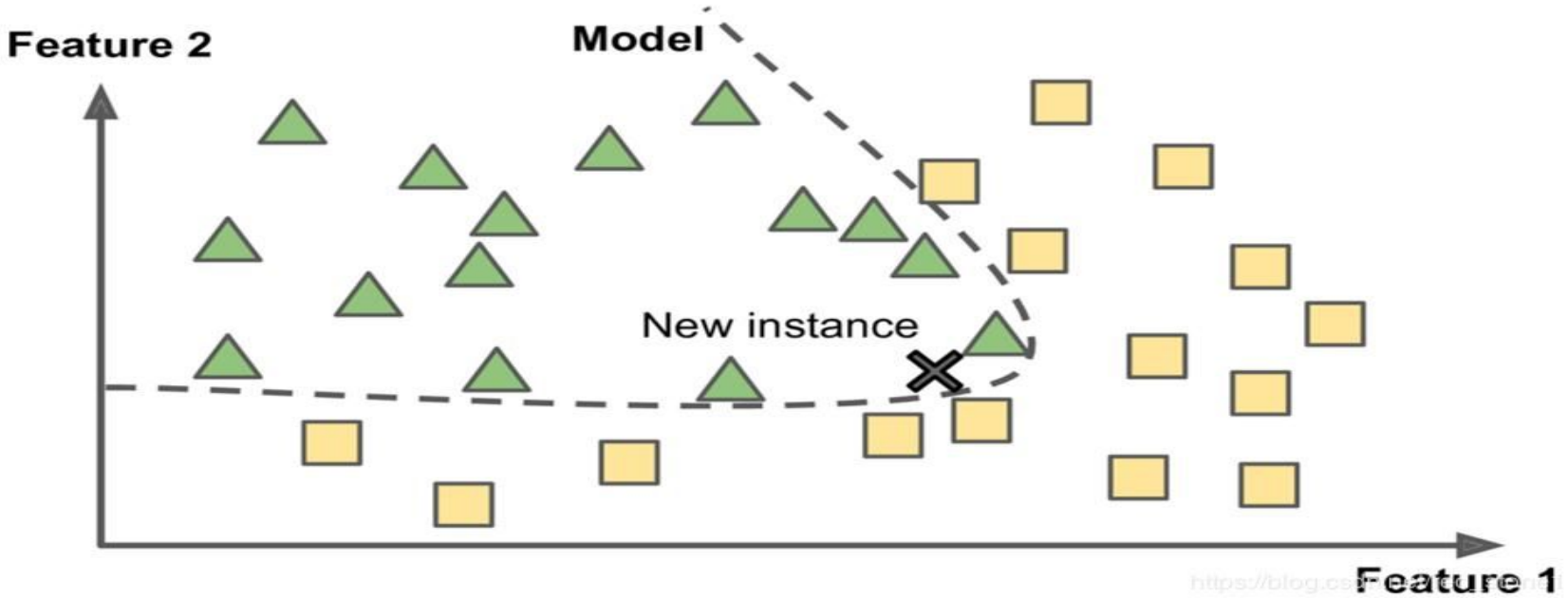
Instance based-algorithms

These algorithms **don't perform explicit generalization**, instead they compare new problem instances with instances seen in training, which have been stored in memory. The system learns the examples.



Machine Learning: How does it work?

Model based-algorithms



"Hands on Machine Learning", Aurélien Géron, O'Reilly

Model based-algorithms: Example

Suppose you want to know if money makes people happy

Do you agree or not?

What do the numbers say?

Machine Learning: How does it work?

Model based-algorithms

Equation 1-1. A simple linear model

$$life\_satisfaction = \theta_0 + \theta_1 \times GDP\_per\_capita$$

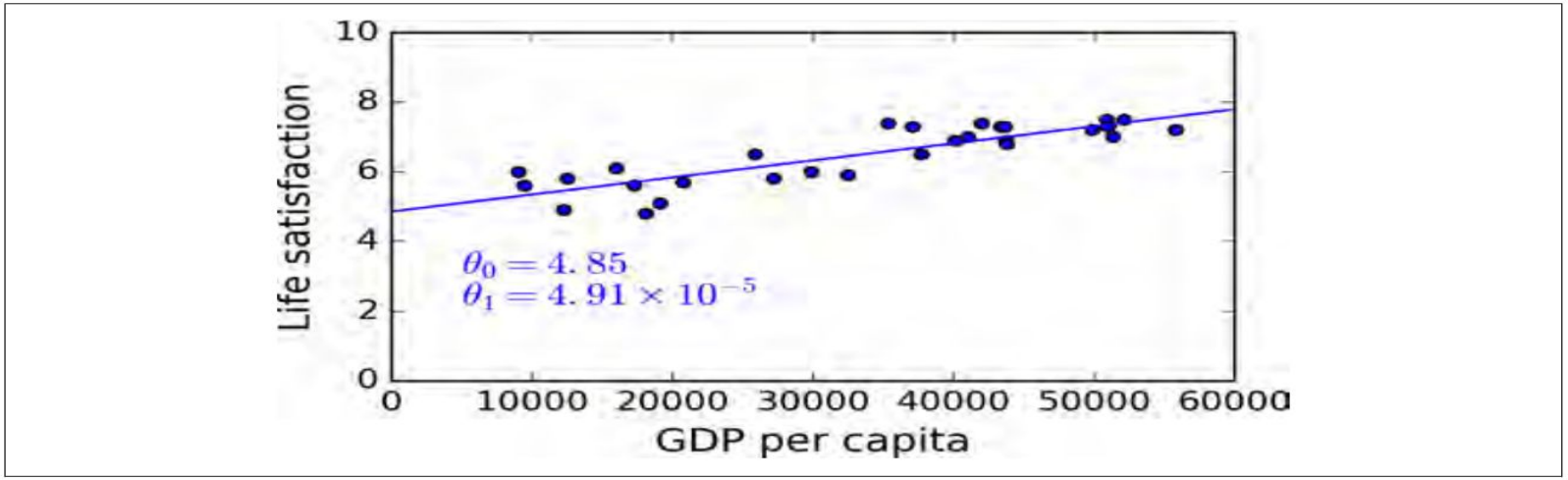


Figure 1-19. The linear model that fits the training data best

You are finally ready to run the model to make predictions. For example, say you want to know how happy Cypriots are, and the OECD data does not have the answer. Fortunately, you can use your model to make a good prediction: you look up Cyprus's GDP per capita, find $22,587, and then apply your model and find that life satisfaction is likely to be somewhere around $4.85 + 22{,}587 \times 4.91 \times 10^{-5} = 5.96$.

"Hands on Machine Learning", Aurélien Géron, O'Reilly

Machine Learning: How does it work?

Model based-algorithms

Equation 1-1. A simple linear model

$$life\_satisfaction = \theta_0 + \theta_1 \times GDP\_per\_capita$$

linear-model.py (~/soft...l/datasets/lifesat) - VIM

```
1  #!/usr/bin/env python
2
3
4  import matplotlib
5  import matplotlib.pyplot as plt
6  import numpy as np
7  import pandas as pd
8  import sklearn
9  import sklearn.linear_model
10 from sklearn.linear_model  import LinearRegression
11
12 # def function "prepare_country_stats"
13
14 def prepare_country_stats(oecd_bli, gdp_per_capita):
15     oecd_bli = oecd_bli[oecd_bli["INEQUALITY"]=="TOT"]
16     oecd_bli = oecd_bli.pivot(index="Country", columns="
   Indicator", values="Value")
17     gdp_per_capita.rename(columns={"2015": "GDP per capi
   ta"}, inplace=True)
18     gdp_per_capita.set_index("Country", inplace=True)
19     full_country_stats = pd.merge(left=oecd_bli, right=g
   dp_per_capita,
20                                   left_index=True, right
   _index=True)
21     full_country_stats.sort_values(by="GDP per capita",
   inplace=True)
22     remove_indices = [0, 1, 6, 8, 33, 34, 35]
23     keep_indices = list(set(range(36)) - set(remove_indi
   ces))
24     return full_country_stats[["GDP per capita", 'Life s
   atisfaction']].iloc[keep_indices]
```

"Hands on Machine Learning", Aurélien Géron, O'Reilly

molmod

Machine Learning: How does it work?

K-Nearest Neighbors Algorithm in Python and Scikit-Learn

- K-nearest neighbors (KNN) algorithm is a type **of supervised machine learning** algorithms;
- KNN is extremely **easy** and yet performs **quite complex** classification tasks;
- Rather, it uses all of the data **for training while classifying a new data** point or instance;
- KNN is a **non-parametric learning algorithm**, which means that it doesn't assume anything about the underlying data;

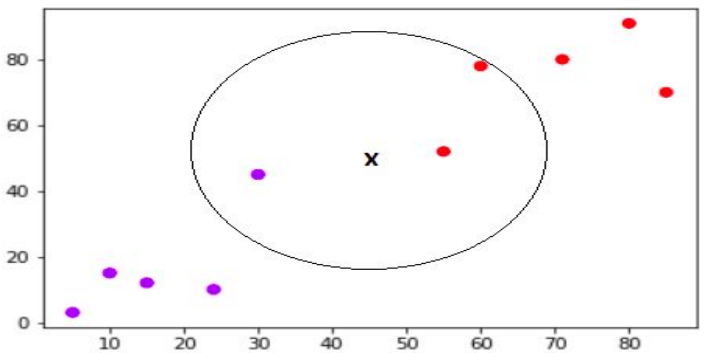
"Hands on Machine Learning", Aurélien Géron, O'Reilly

molmod

Machine Learning: How does it work?

K-Nearest Neighbors Algorithm in Python and Scikit-Learn

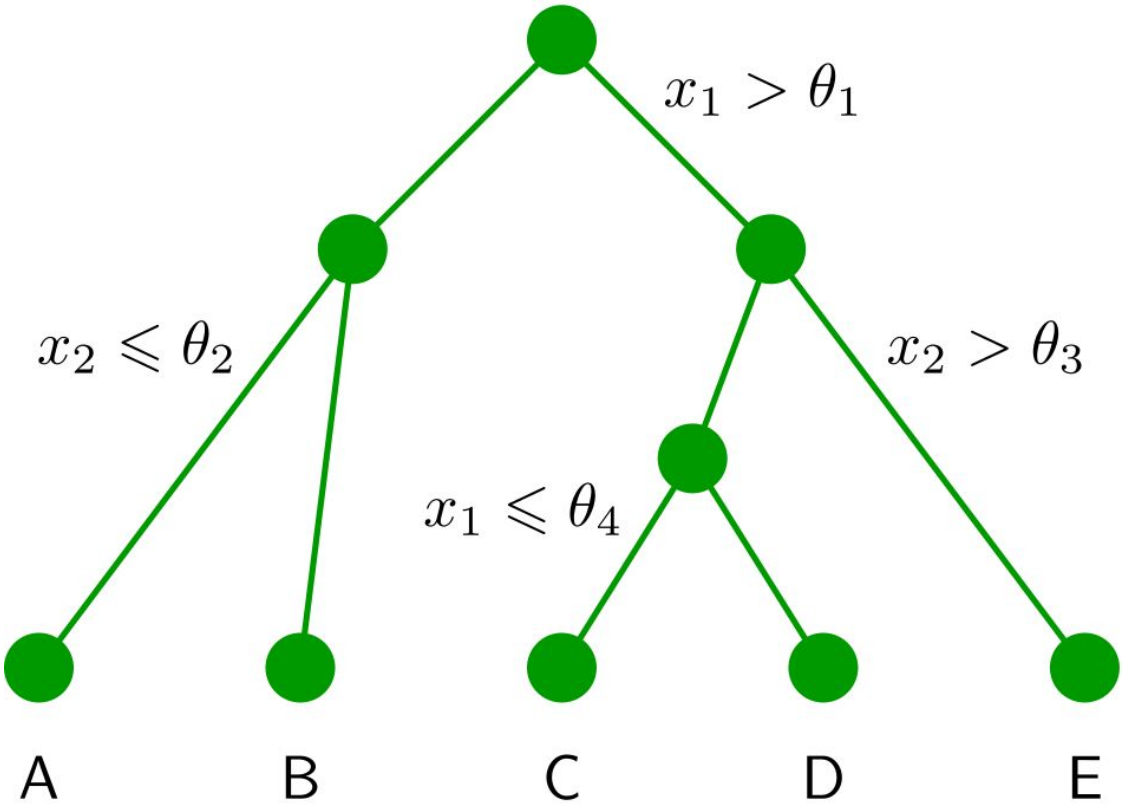
The intuition behind the KNN algorithm is one of the simplest of all the supervised machine learning algorithms. It simply calculates the distance of a new data point to all other training data points. The distance can be of any type e.g **Euclidean or Manhattan** etc. It then selects the K-nearest data points, where K can be any integer. Finally it assigns the data point to the class to which the majority of the K data points belong.



https://stackabuse.com/k-nearest-neighbors-algorithm-in-python-and-scikit-learn/

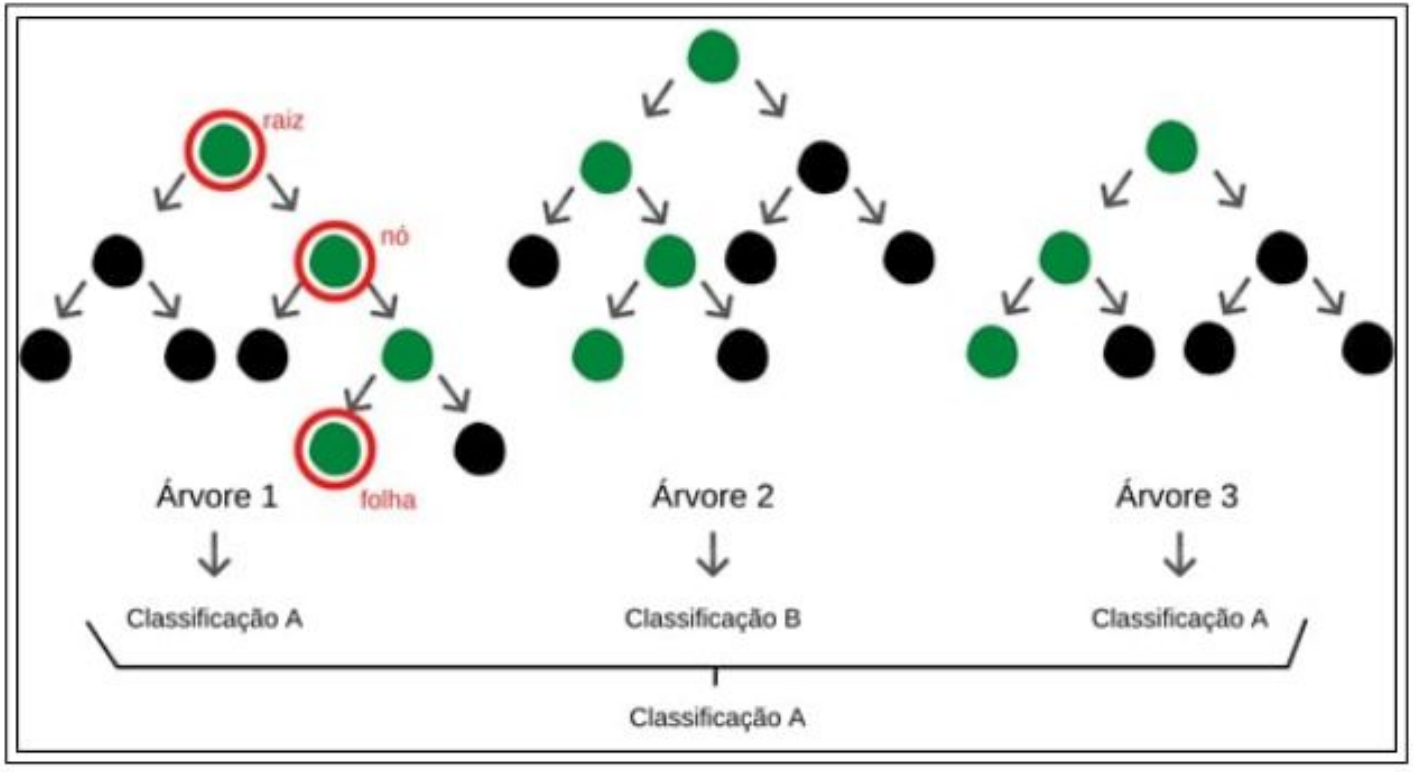
Machine Learning: How does it work?

Decision Tree



Machine Learning: How does it work?

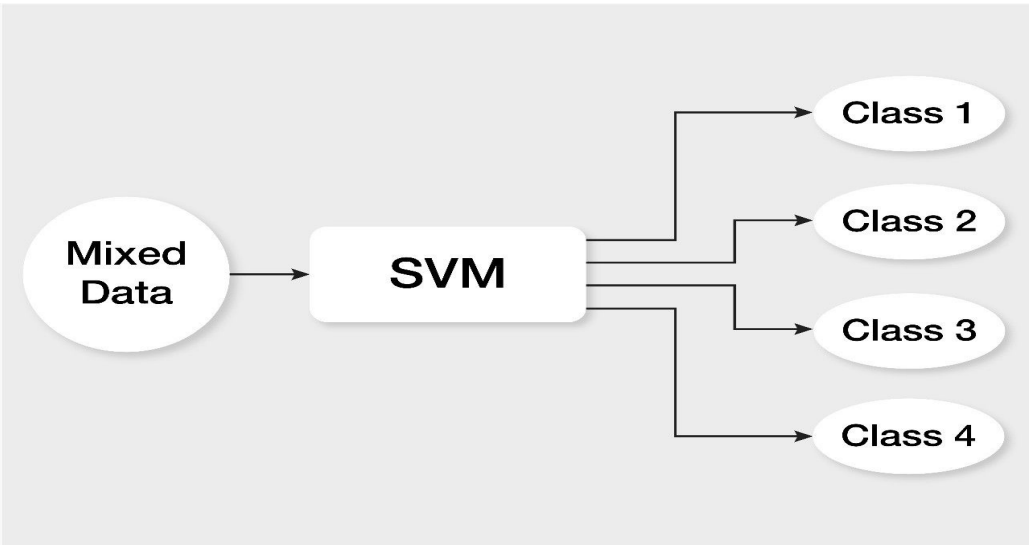
Random Forest



Machine Learning: How does it work?

Support Vector Machines

SVM or **Support Vector Machine** is a linear model for classification and regression problems. It can solve linear and non-linear problems and work well for many practical problems. The idea of SVM is simple: The algorithm creates a line or a hyperplane which separates the data into classes.

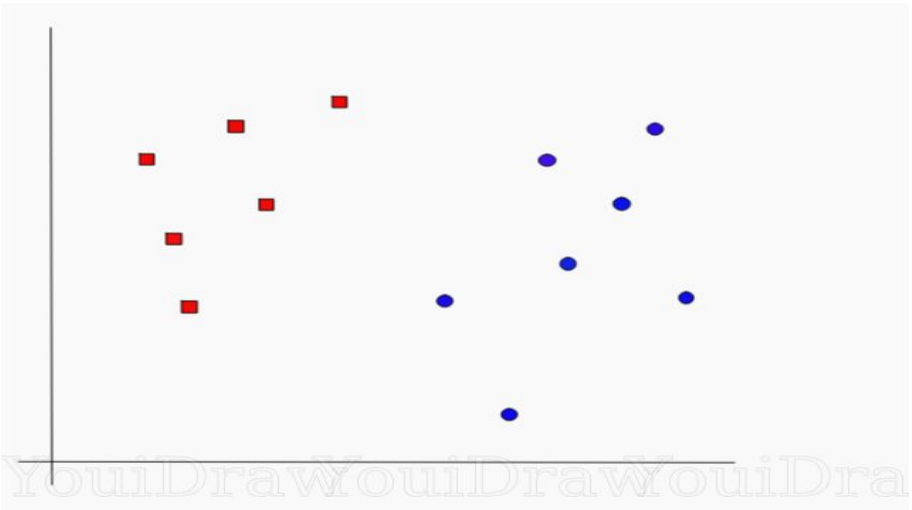


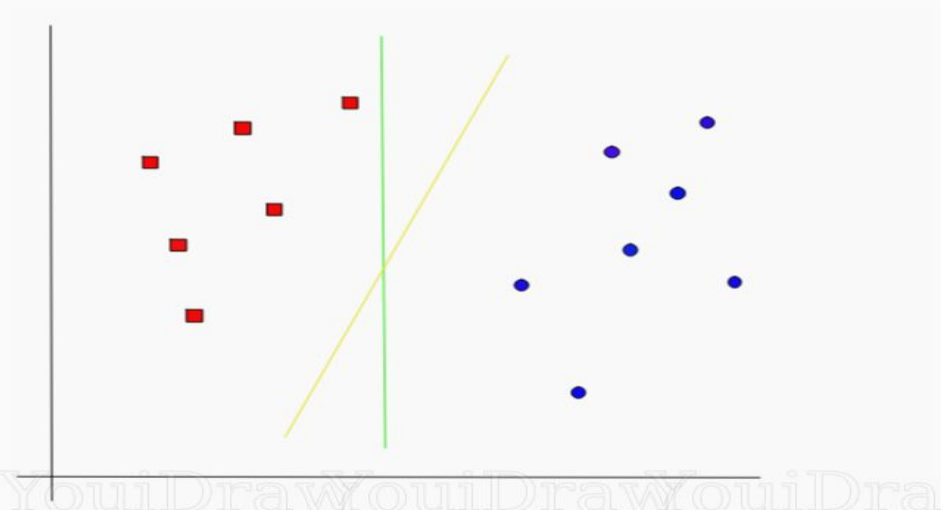
https://www.quantib.com/the-ultimate-guide-to-ai-in-radiology

Machine Learning: How does it work?

Support Vector Machines

We find the points closest to the line from both the classes.These points are called support vectors. Now, we **compute the distance between the line and the support vectors**. This distance is called the margin. Our goal is **to maximize the margin**. The hyperplane for which the margin is maximum is the optimal hyperplane.



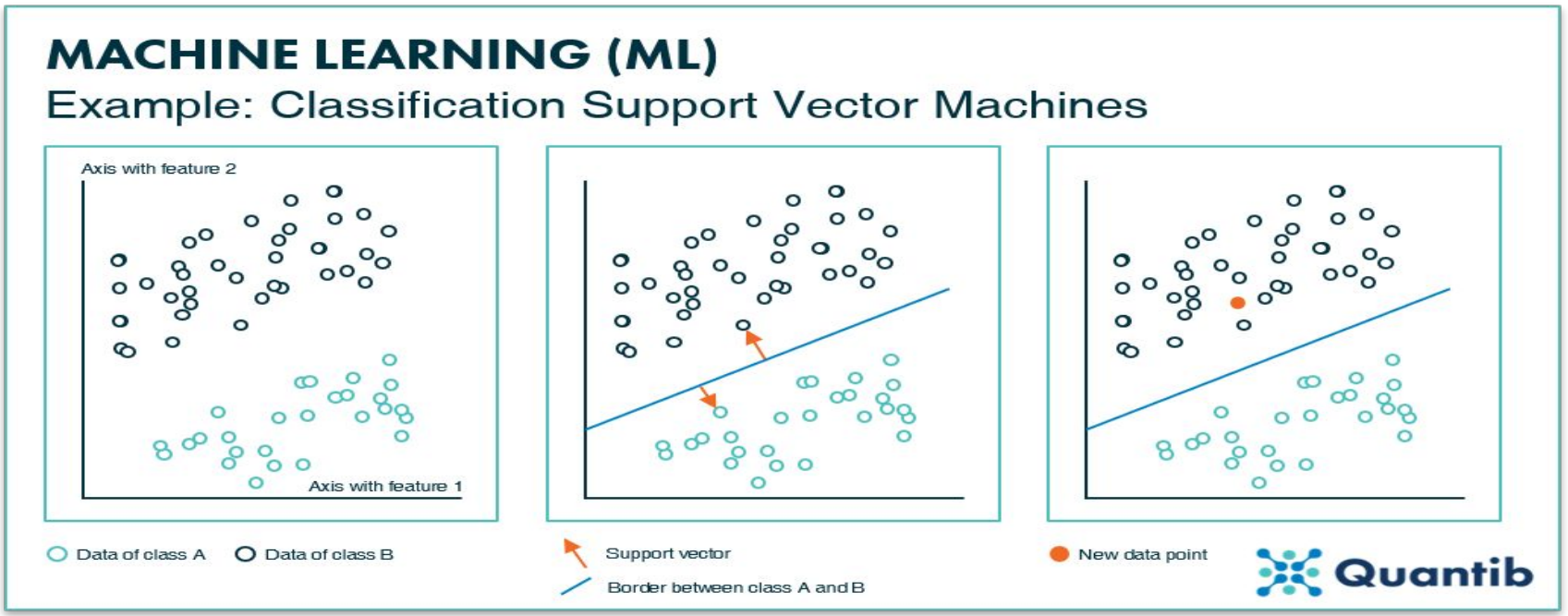


https://www.quantib.com/the-ultimate-guide-to-ai-in-radiology

Machine Learning: How does it work?

Support Vector Machines

All features combined can be represented in feature space. A visual representation of feature space (showing features in a graph) can help to get an overview of all feature values. The simplest example is when your dataset has two features. You visualize this by drawing a graph with one feature on the x-axis and the other feature on the y-axis.



https://www.quantib.com/the-ultimate-guide-to-ai-in-radiology

Machine Learning: How does it work?

Support Vector Machines

```
In [2]: import numpy as np

In [3]: X = np.array([[-1,-1],[-2,-1],[1,1],[2,1]])

In [4]: Y = np.array([1,1,2,2])

In [5]: # Here we have our points in X and the classes they
   ...: belong to in Y

In [6]: # Now we need to train the SVM model with the above
   ...: dataset

In [7]: from sklearn.svm import SVC

In [8]: clf = SVC(kernel='linear')

In [9]: clf.fit(X,Y)
Out[9]:
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='auto_dep
recated',
    kernel='linear', max_iter=-1, probability=False, random_
state=None,
    shrinking=True, tol=0.001, verbose=False)

In [10]: # To predict the class of new dataset

In [11]: prediction = clf.predict([[0,6]])

In [12]: print(prediction)
[2]

In [13]: prediction = clf.predict([[3,3]])
```

https://towardsdatascience.com/

molmod

Machine Learning: How does it work?

K-Means Clustering

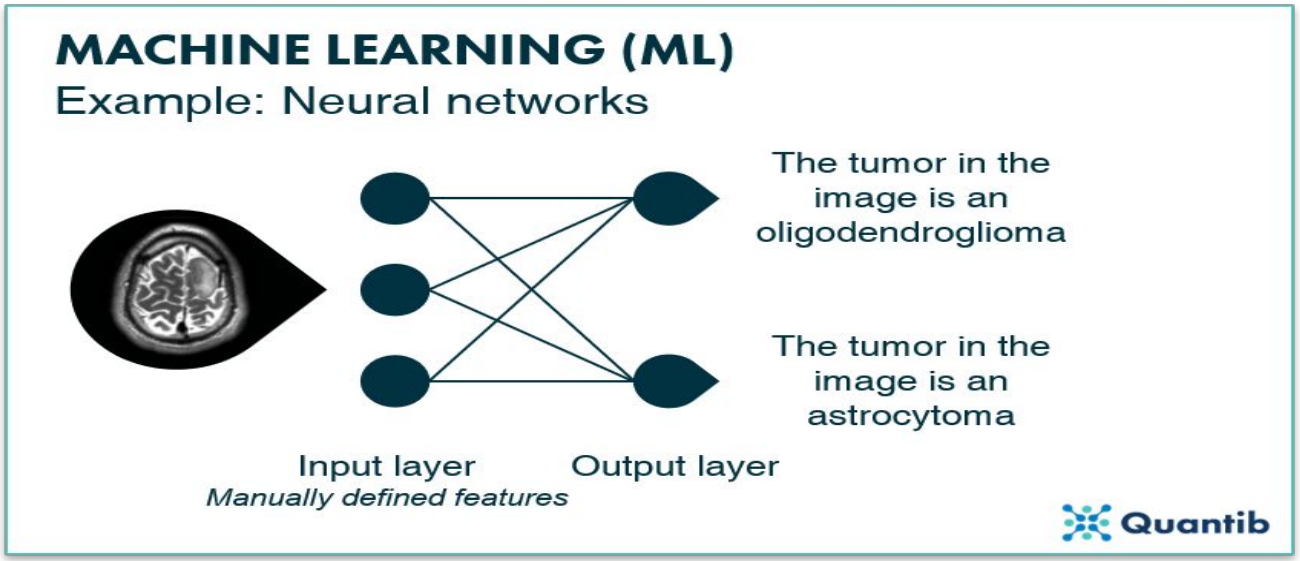
Clustering is quite similar to classification in the sense that these methods group input data into different classes based on predefined features. However, clustering techniques are typically used when no ground-truth labels are available for the different classes. Thus, clustering algorithms define their own grouping system (or “labeling”, if you will) to classify the input data. Hence it should be considered an **unsupervised learning method**.

https://towardsdatascience.com/

Machine Learning: How does it work?

(Artificial) Neural Networks

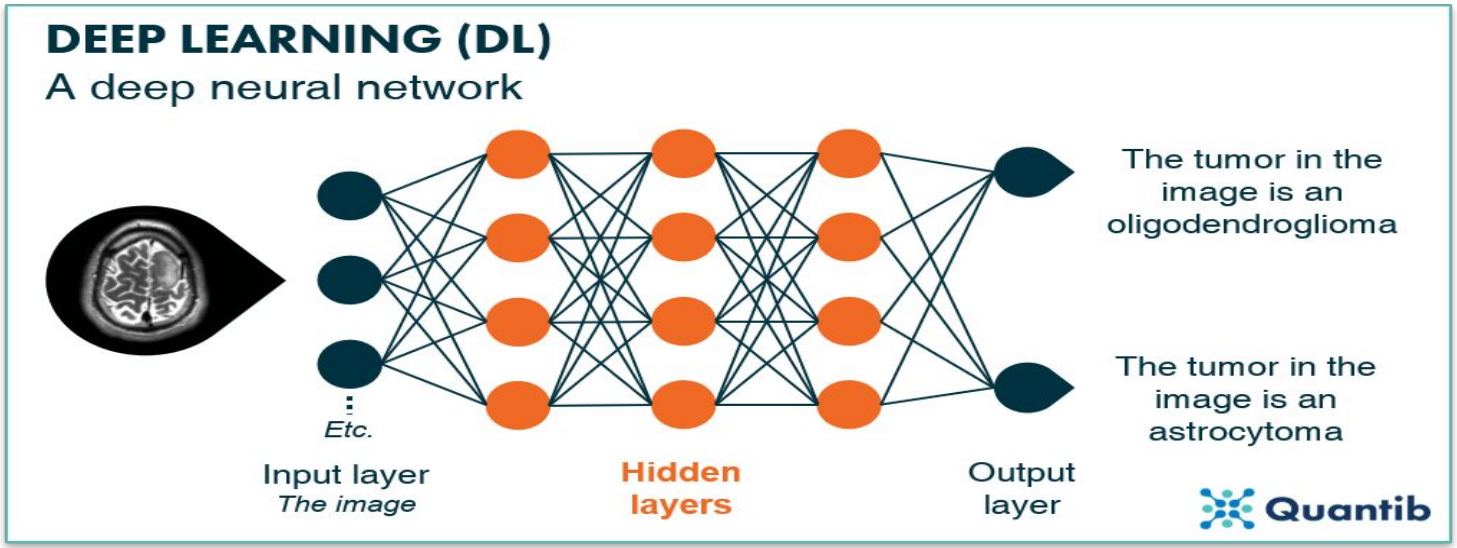
Neural networks are also part of the machine learning realm. It is a specific group of methods which solve classification problems, but they are also suited to function as a **regression or clustering technique** or they can perform segmentation tasks.



Machine Learning: How does it work?

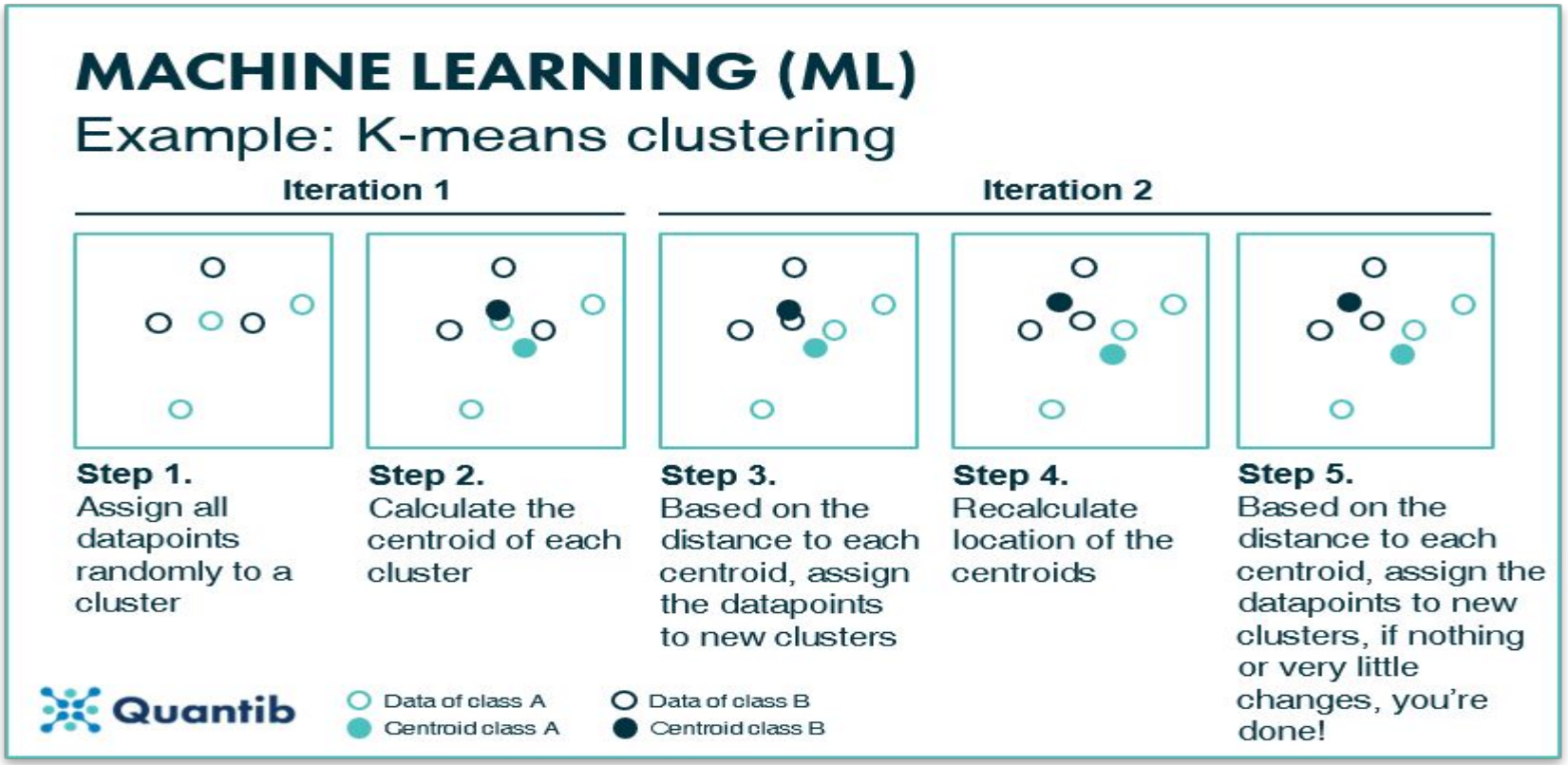
Deep Learning

Deep learning is a subset of machine learning with the main differentiating factor being that deep learning uses “deep neural networks”, whereas machine learning comprises a much broader set of techniques. **Deep neural networks** are similar to the simple network described previously. However, deep networks have **hidden layers** between the input and the output layer to refine the calculations and hence the predictions



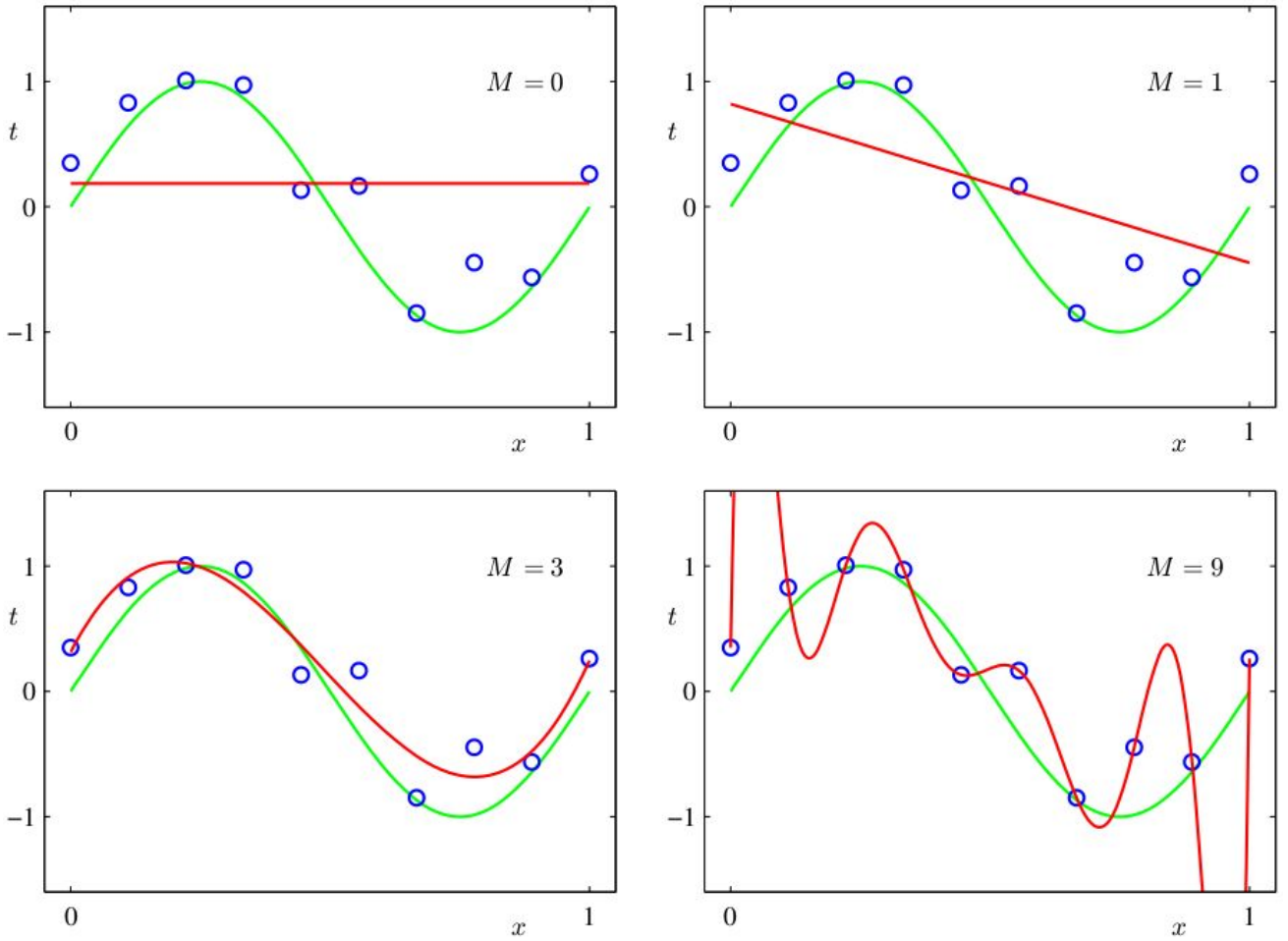
Machine Learning: How does it work?

K-Means Clustering

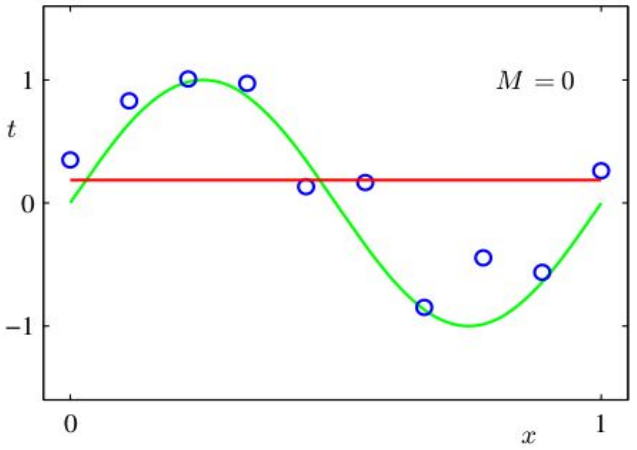


https://towardsdatascience.com/

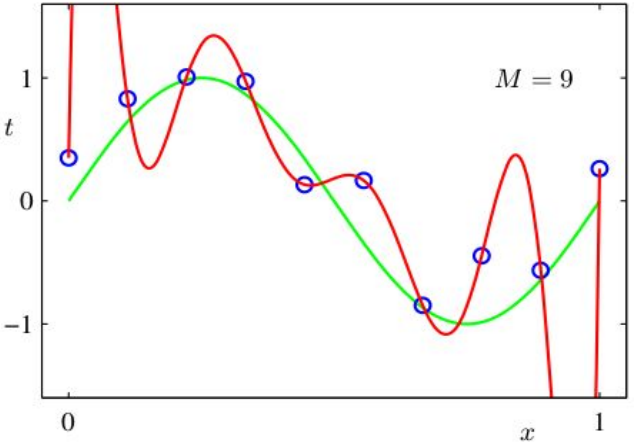
Machine Learning: Model Selection



Machine Learning: Model Selection

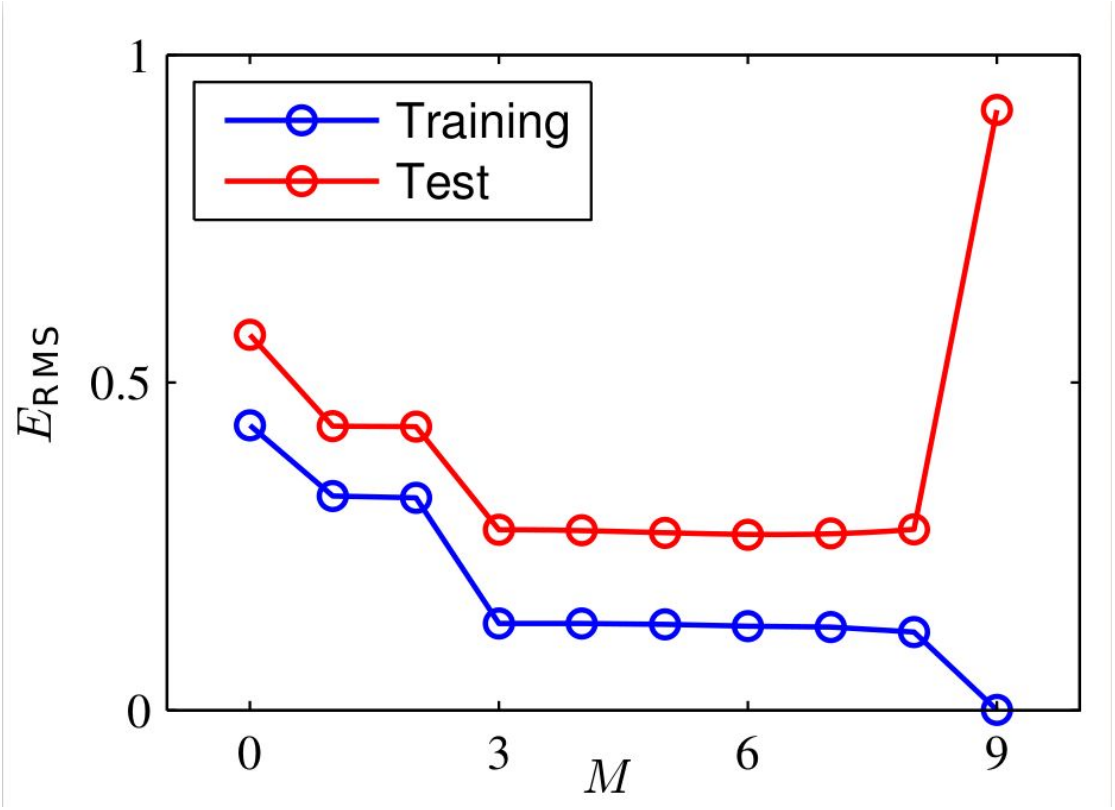


"*underfitting*"
- high bias

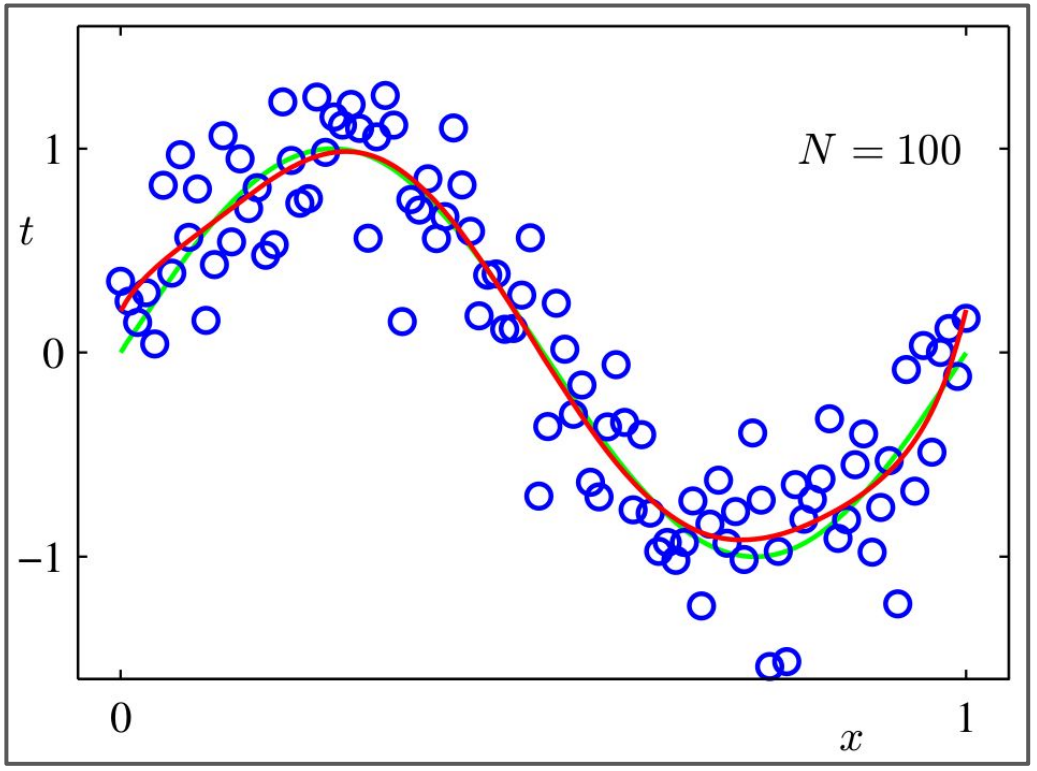


"*overfitting*"
- high variance

Machine Learning: Model Selection



Machine Learning: Model Selection



Machine Learning: Model Selection

Challenges of Machine Learning

- Insufficient Quality of Training Data
- Nonrepresentative Training Data
- Poor Quality Data
- Irrelevant Features
- Overfitting the Training Data
- Underfitting the Training Data

Machine Learning: Principles

- The simplest model that fits the data is also the most plausible (Occam's Razor)
- If the data is samples in a biased way, learning will produce a similarly biased outcome
- If a data set has affected any step in the learning process, its ability to assess the outcome has been compromised

Additional Material

Install Tools: Python and Jupyter Notebook

```
sudo apt update && sudo apt upgrade
```

```
sudo apt install python3
```

```
sudo apt install python3-pip
```

```
sudo pip3 install --upgrade pip
```

```
pip install jupyter
```

```
jupyter notebook
```

https://linux.how2shout.com/how-to-install-jupyter-on-ubuntu-20-04-lts-linux/

Install Tools: Scikit-learn

```
pip install scikit-learn --user
```

```
python3
```

```
import sklearn
```

```
print(sklearn._ _version_ _)
```

https://linux.how2shout.com/installing-scikit-learn-sklearn-using-pip-a-step-by-step-guide/

Public DataSets

- UC Irvine Machine Learning Repository (http://archive.ics.uci.edu/ml/index.php): We currently maintain 497 data sets as a service to the machine learning community. You may view all data sets through our searchable interface

- Kaggle repository (https://www.kaggle.com/datasets):

- Amazon AWS repository (https://registry.opendata.aws/):
  META PORTAL: (They list open repositories)

  http://dataportals.org/

  http://opendatamonitor.eu/

  http://quandl.com/

We can import “numpy” as:

```
IPython: MethodsFQ/PYTHON

In [2]: import numpy as np

In [3]: from matplotlib import pyplot as plt
```

Or:

```
IPython: MethodsFQ/PYTHON

In [5]: import matplotlib.pyplot as plt
```

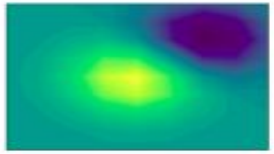
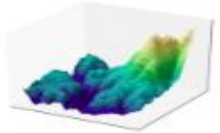
From matplotlib webpage: “matplotlib tries to make easy things easy and hard things possible. (...) For simple plotting the pyplot interface provides a MATLAB-like interface, particularly when combined with IPython.”

matplotlib Version 3.1.1

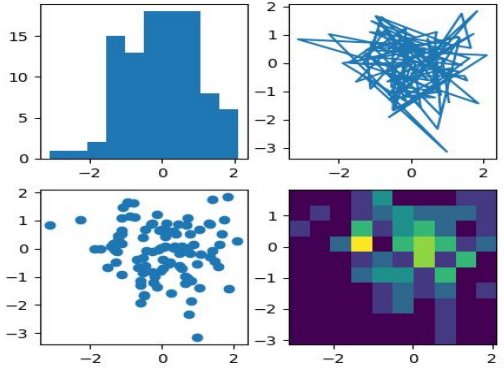
home | examples | tutorials | API | contents »

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter notebook, web application servers, and four graphical user interface toolkits.



```
import matplotlib.pyplot as plt
import numpy as np

np.random.seed(19680801)
data = np.random.randn(2, 100)

fig, axs = plt.subplots(2, 2,
figsize=(5, 5))
axs[0, 0].hist(data[0])
axs[1, 0].scatter(data[0], data[1])
```



How to plot a function *f(x)*?

$$y = f(x)$$

$$x = [a, b]$$

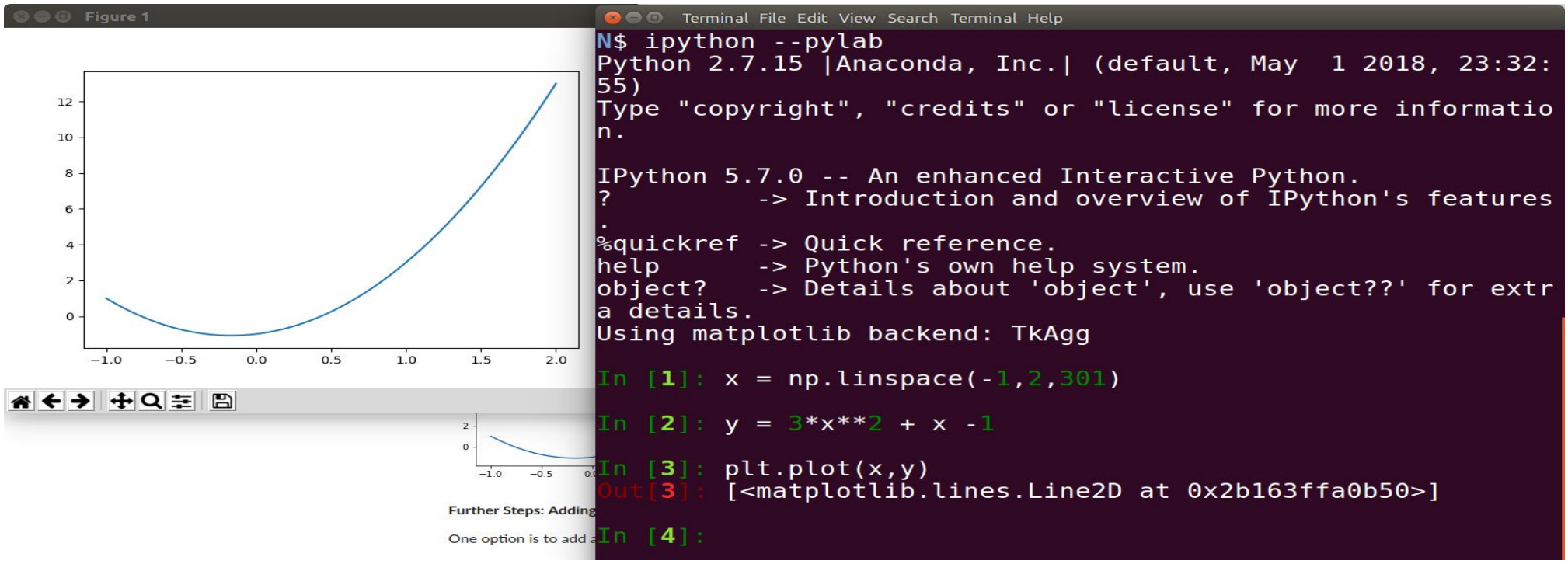
Thinking in the algorithm ...

1. We need to define the range for x
2. We use the plt.plot command in which the function is defined

Example: $f(x) = 3x^2 + x - 1$

$$x = [-1, 2]$$

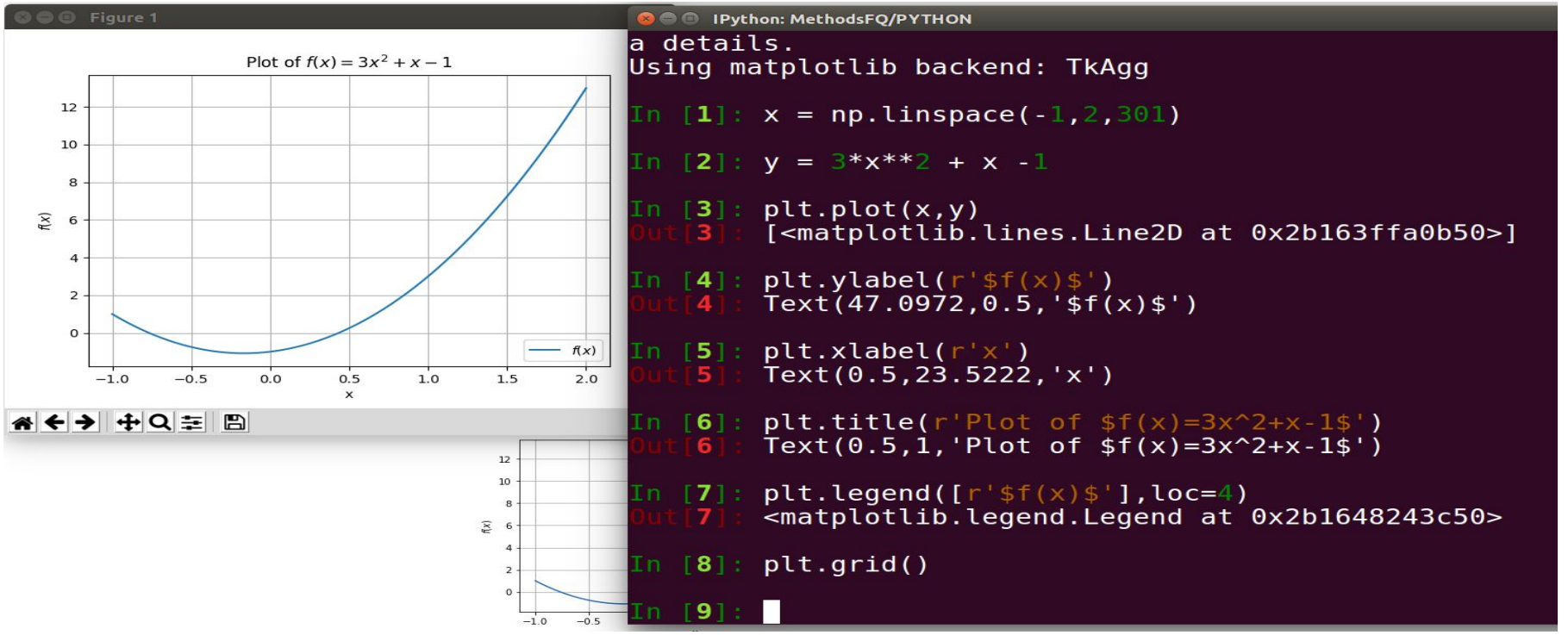
Basic Plotting in Python using Matplotlib



and if you want to plot with **axes labels**, **legend**, **title**, **grid**, and so on?

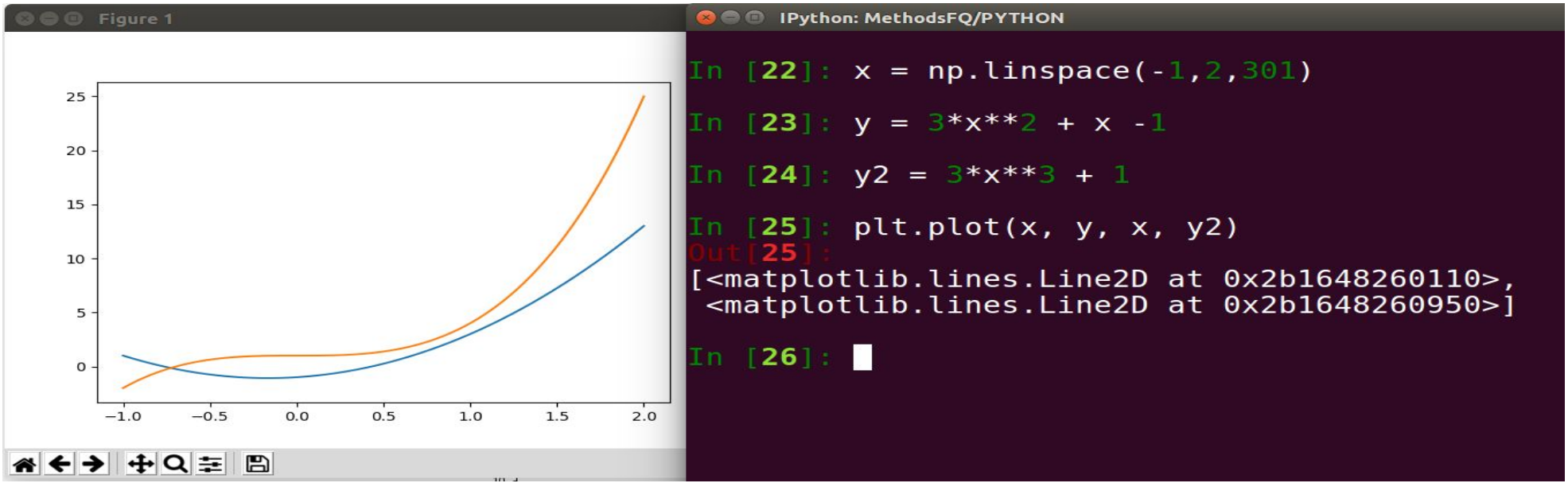
Basic Plotting in Python using Matplotlib

Then we can follow this receipt here:



Basic Plotting in Python using Matplotlib

Plotting multiple functions using the same range of x values:



To create legends in a different local, just typing:

```
In [26]: plt.legend([r'$f(x)$',r'$g(x)$'], loc=2);
```

Basic Plotting in Python using Matplotlib

It is also possible edit the figure using parameters such as follows:

The *plt.figure()* command also lets you set other parameters related to the figure. In most of the other modules and examples, we have set common figure parameters in the beginning of the module/example, by using the following lines of code:

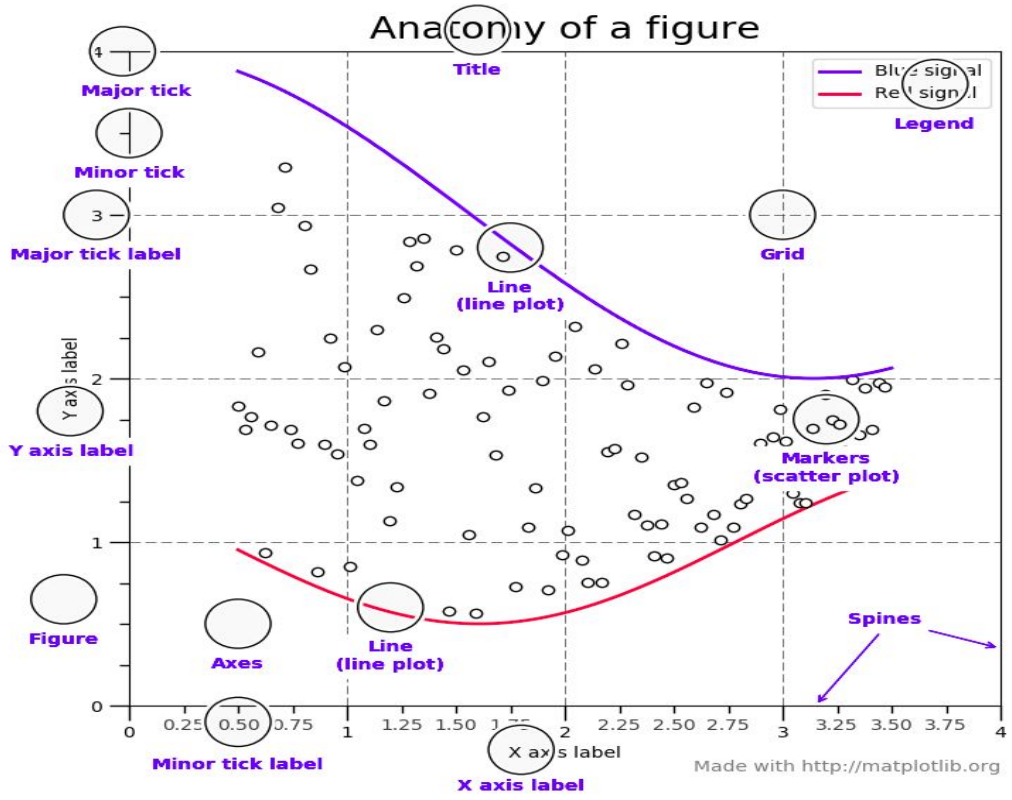
In [11]:
```
from matplotlib import rc
# Set common figure parameters:
newparams = {'axes.labelsize': 11, 'axes.linewidth': 1, 'savefig.dpi': 300,
             'lines.linewidth': 1.0, 'figure.figsize': (8, 3),
             'ytick.labelsize': 10, 'xtick.labelsize': 10,
             'ytick.major.pad': 5, 'xtick.major.pad': 5,}
plt.rcParams.update(newparams)
```

plt.figure() module in matplotlib can manage the figure. The command help(plt.figure) opens a short guide for the tool.

Pyplot's state-machine environment behaves similarly to MATLAB and should be most familiar to users with MATLAB experience.

Basic Plotting in Python using Matplotlib

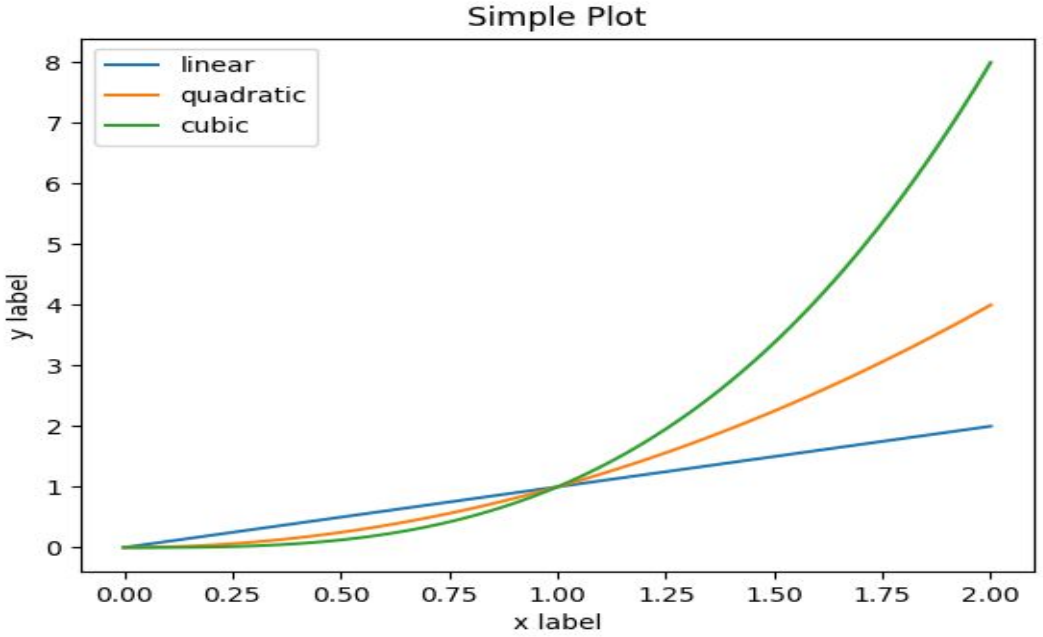
Resuming the detailed description of a figure by matplotlib.pyplot module



Figure

The **whole** figure. The figure keeps track of all the child Axes, a smattering of 'special' artists (titles, figure legends, etc), and the **canvas**. (Don't worry too much about the canvas, it is crucial as it is the object that actually does the drawing to get you your plot, but as the user it is more-or-less invisible to you). A figure can have any number of Axes, but to be useful should have at least one.

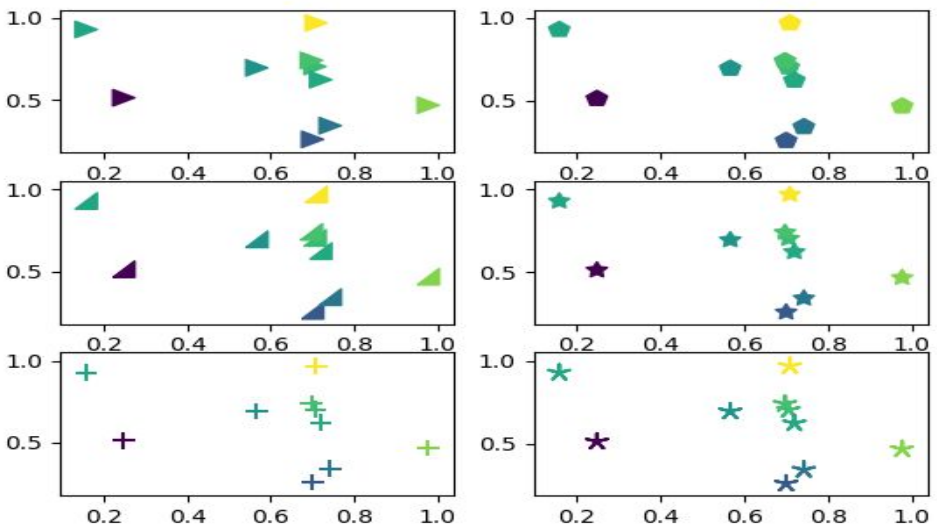
It is also possible plotting a graph directly using **plt.plot module**, like this:




```
x = np.linspace(0, 2, 100)

plt.plot(x, x, label='linear')
plt.plot(x, x**2, label='quadratic')
plt.plot(x, x**3, label='cubic')

plt.xlabel('x label')
plt.ylabel('y label')
plt.title("Simple Plot")

plt.legend()
plt.show()
```

Basic Plotting in Python using Matplotlib

Plotting multiple graphs in the same figure

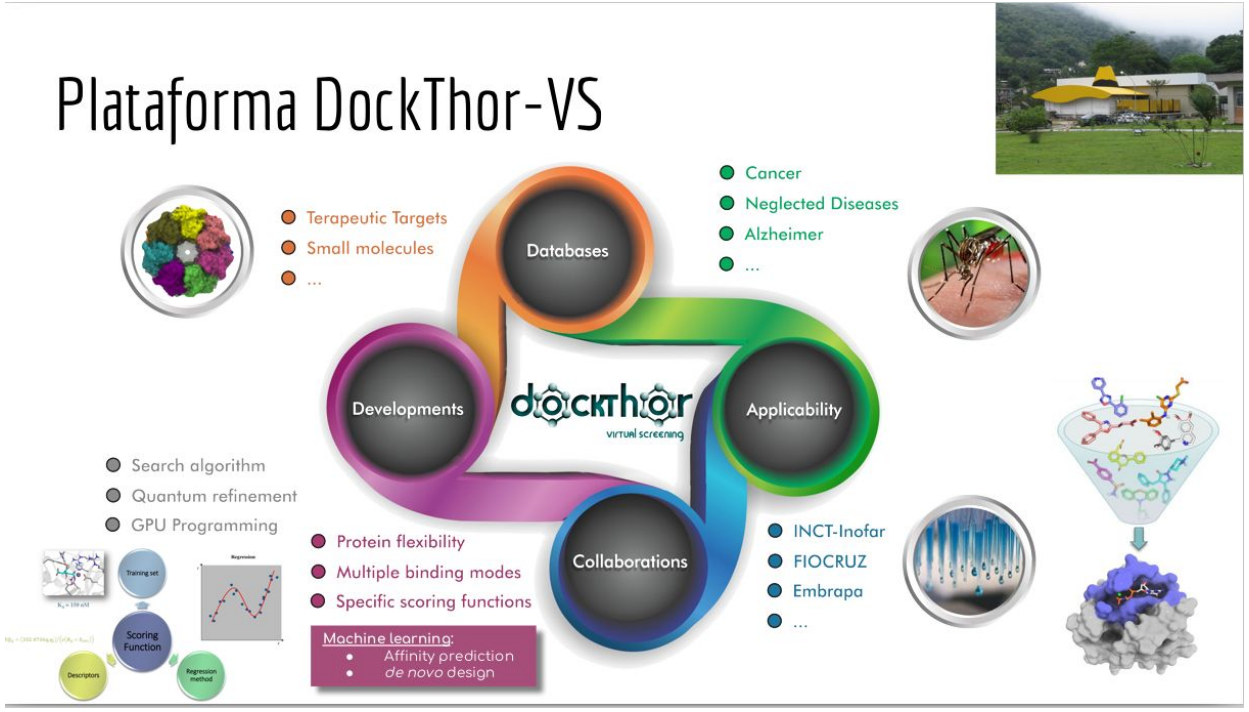


scatter_star_poly.py (~/WORK/...r-2016-2/MethodsFQ/PYTHON) - VIM

```
"""
==================
Scatter Star Poly
==================

Create multiple scatter plots with different
star symbols.

"""
import numpy as np
import matplotlib.pyplot as plt

# Fixing random state for reproducibility
np.random.seed(19680801)


x = np.random.rand(10)
y = np.random.rand(10)
z = np.sqrt(x**2 + y**2)

plt.subplot(321)
plt.scatter(x, y, s=80, c=z, marker=">")

plt.subplot(322)
plt.scatter(x, y, s=80, c=z, marker=(5, 0))

verts = np.array([[-1, -1], [1, -1], [1, 1], [-1, -1]])
plt.subplot(323)
plt.scatter(x, y, s=80, c=z, marker=verts)
                                                    1,1
```

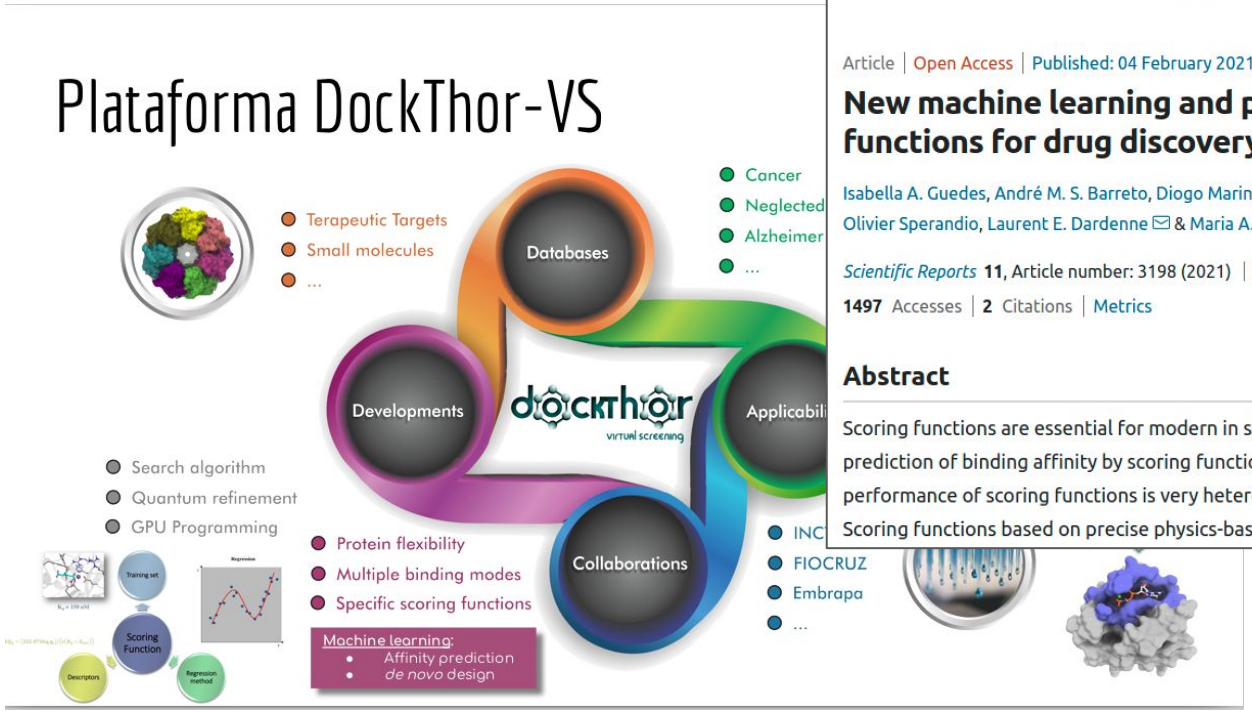
Machine Learning: Examples

DockThor-VS



www.dockthor.lncc.br

Machine Learning: Examples



www.dockthor.lncc.br

scientific reports

Explore content · Journal information · Publish with us

nature › scientific reports › articles › article

Article | Open Access | Published: 04 February 2021

New machine learning and physics-based scoring functions for drug discovery

Isabella A. Guedes, André M. S. Barreto, Diogo Marinho, Eduardo Krempser, Mélaine A. Kuenemann, Olivier Sperandio, Laurent E. Dardenne & Maria A. Miteva

Scientific Reports **11**, Article number: 3198 (2021) | Cite this article

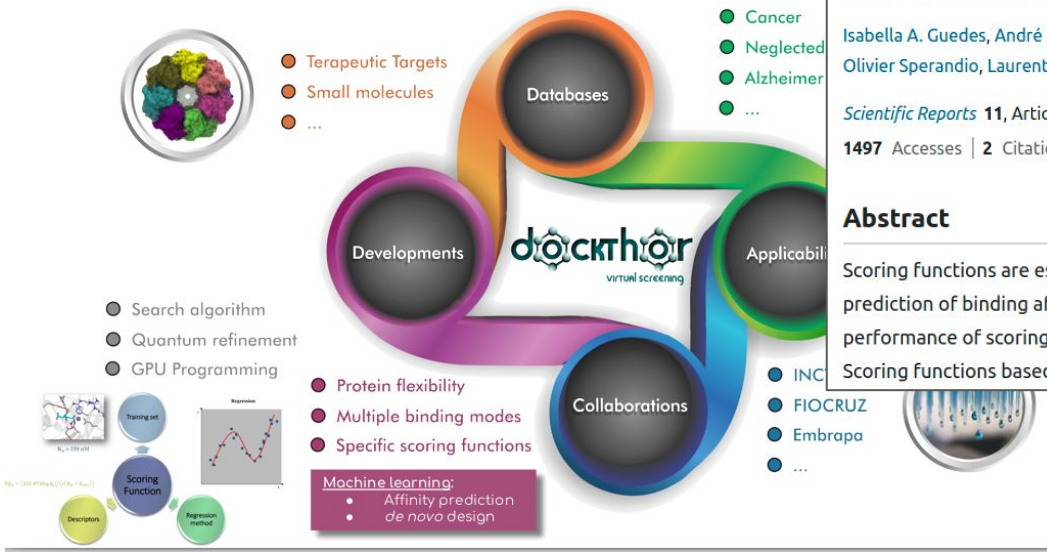
1497 Accesses | **2** Citations | Metrics

Abstract

Scoring functions are essential for modern in silico drug discovery. However, the accurate prediction of binding affinity by scoring functions remains a challenging task. The performance of scoring functions is very heterogeneous across different target classes. Scoring functions based on precise physics-based descriptors better representing protein–

Machine Learning: Examples

Plataforma DockThor-VS



www.dockthor.lncc.br

scientific reports

Explore content | Journal information | Publish with us

nature > scientific reports > articles > article

Article | Open Access | Published: 04 February 2021

New machine learning and physics-based scoring functions for drug discovery

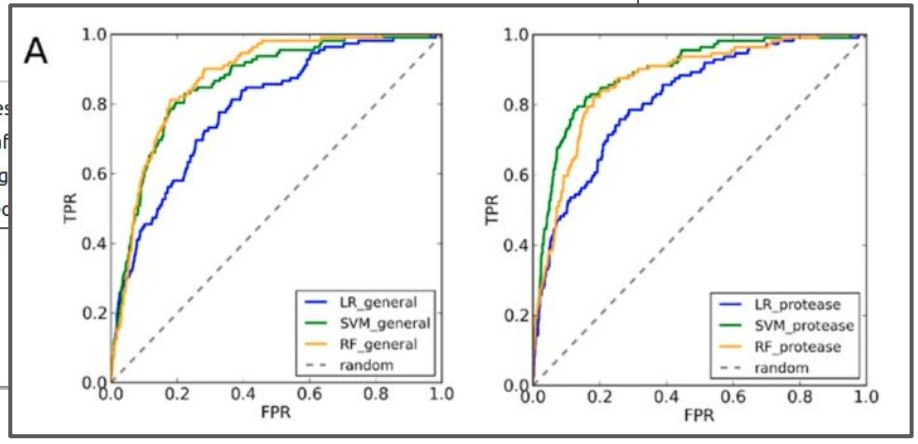
Isabella A. Guedes, André M. S. Barreto, Diogo Marinho, Eduardo Krempser, Mélaine A. Kuenemann, Olivier Sperandio, Laurent E. Dardenne & Maria A. Miteva

Scientific Reports **11**, Article number: 3198 (2021) | Cite this article

1497 Accesses | **2** Citations | Metrics

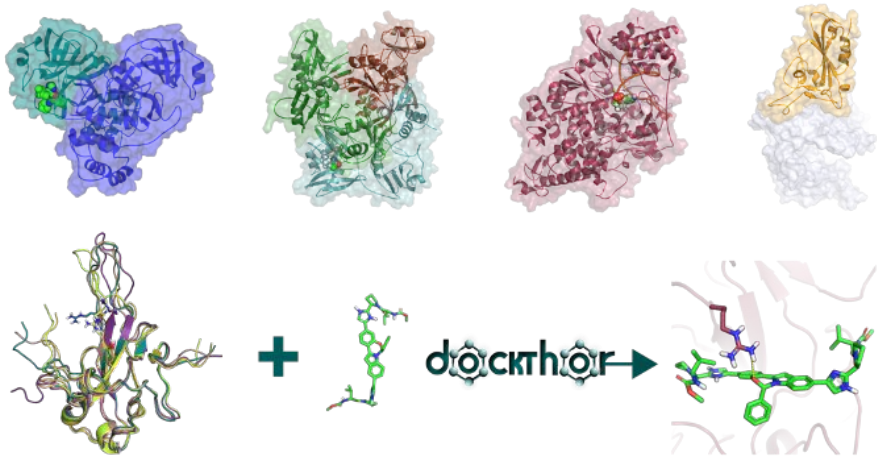
Abstract

Scoring functions are es
prediction of binding af
performance of scoring
Scoring functions based



Machine Learning: Examples

DockThor-VS | COVID-19



scientific reports

Drug design and repurposing with DockThor-VS web server focusing on SARS-CoV-2 therapeutic targets and their non-synonym variants

Isabella A. Guedes, Leon S. C. Costa, Karina B. dos Santos, Ana L. M. Karl, Gregório K. Rocha, Iury M. Teixeira, Marcelo M. Galheigo, Vivian Medeiros, Eduardo Krempser, Fábio L. Custódio, Helio J. C. Barbosa, Marisa F. Nicolás ✉ & Laurent E. Dardenne ✉

Scientific Reports **11**, Article number: 5543 (2021) | Cite this article

5326 Accesses | **11** Citations | Metrics

Machine Learning: Examples

Inhibiting the HIV-1 Integrase

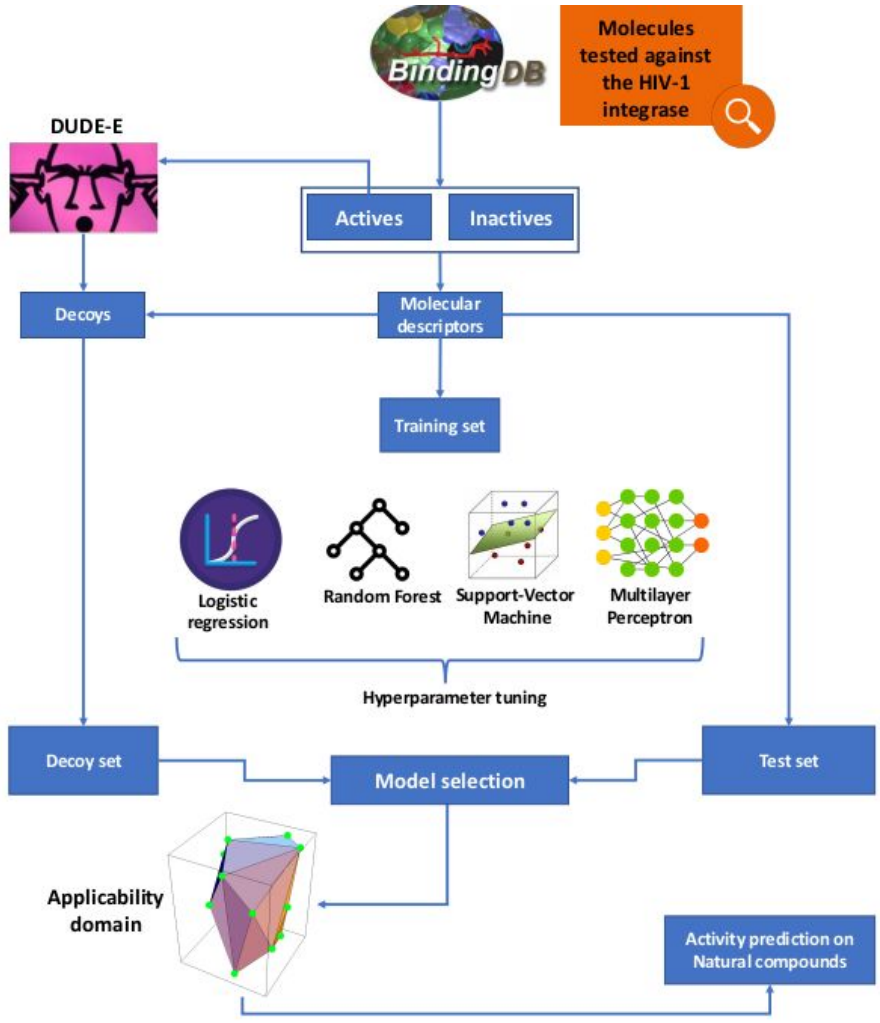
ORIGINAL RESEARCH article

Front. Drug Discov., 21 October 2022
Sec. Anti-Infective Agents
Volume 2 - 2022 | https://doi.org/10.3389/fddsv.2022.954911

This article is part of the Research Topic
In Silico Discovery of Antimicrobials
View all 3 Articles >

A machine learning-based virtual screening for natural compounds capable of inhibiting the HIV-1 integrase

Lucas A. Machado[1]*, Eduardo Krempser[2] and Ana Carolina Ramos Guimarães[3]



Machine Learning: Examples

Industrial

https://www.infomoney.com.br/mercados/o-boticario-desenvolve-primeiro-perfume-do-mundo-usando-inteligencia-artificial/