S-PLUS Southern Photometric Local Universe Survey

Introduction to Machine Learning

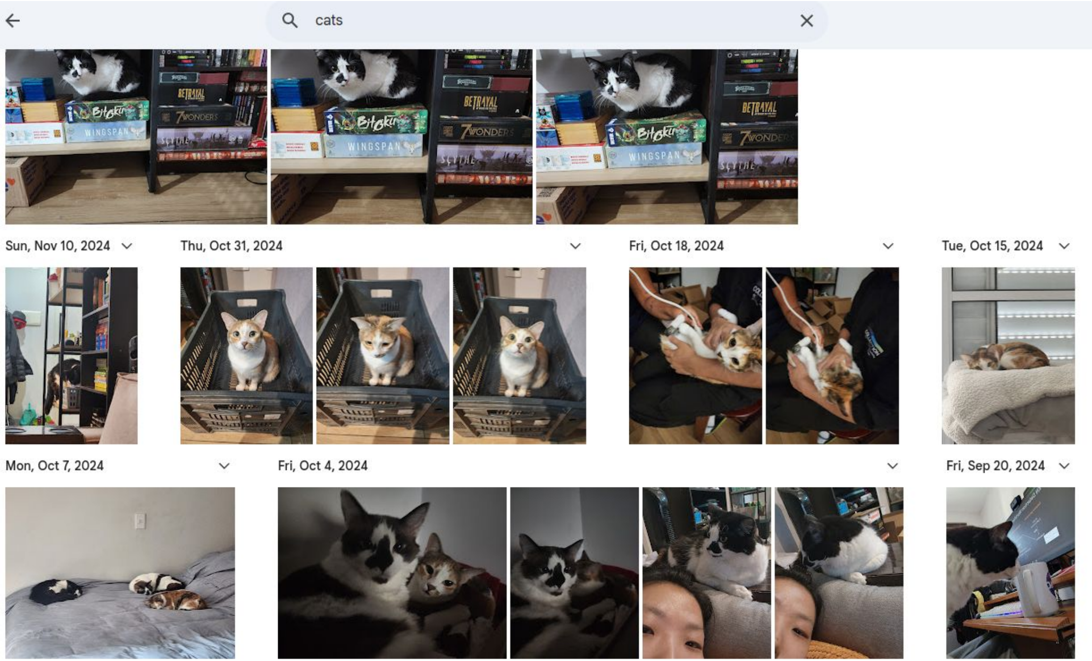
Lilianne Nakazono (lilianne.nakazono@gmail.com)
Instituto de Física, Universidade de São Paulo
(In May: Technology Specialist at Observatório Nacional, Rio de Janeiro)

XI LAPIS

07 April 2025

Applications and different types of learning

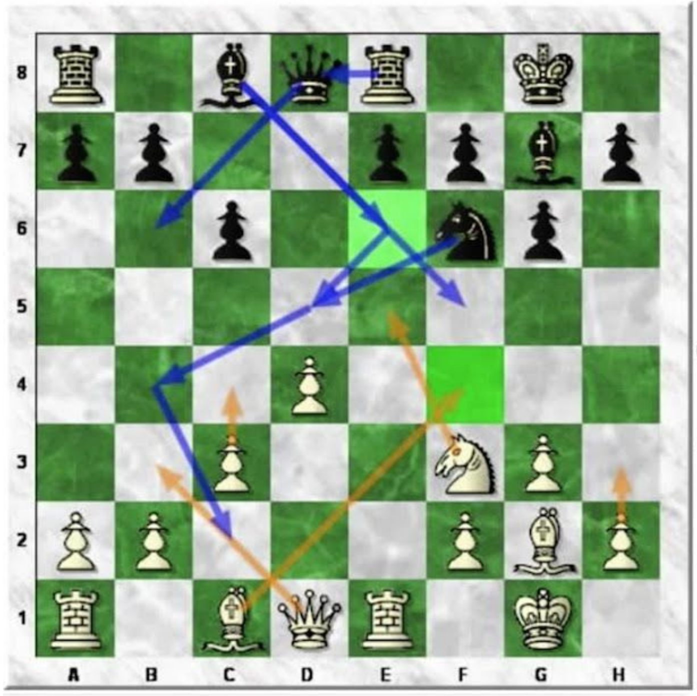
Image classification ⇒ supervised learning

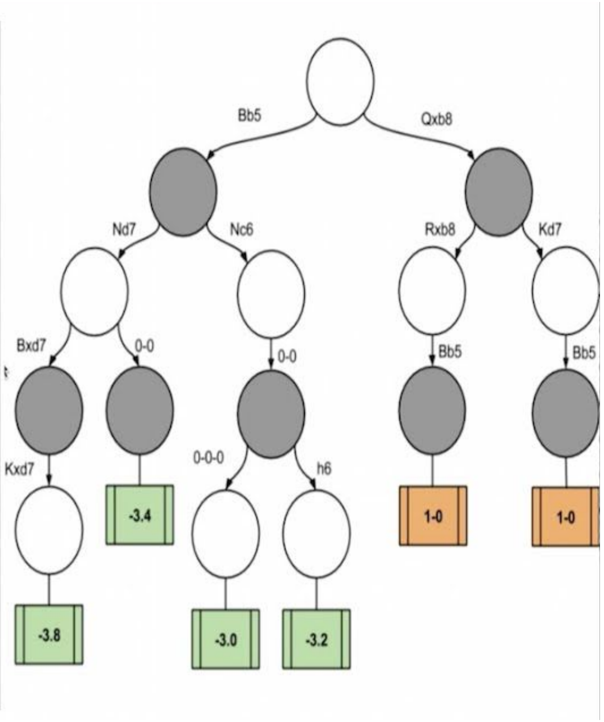


Source: my own Google Photos...

Applications and different types of learning

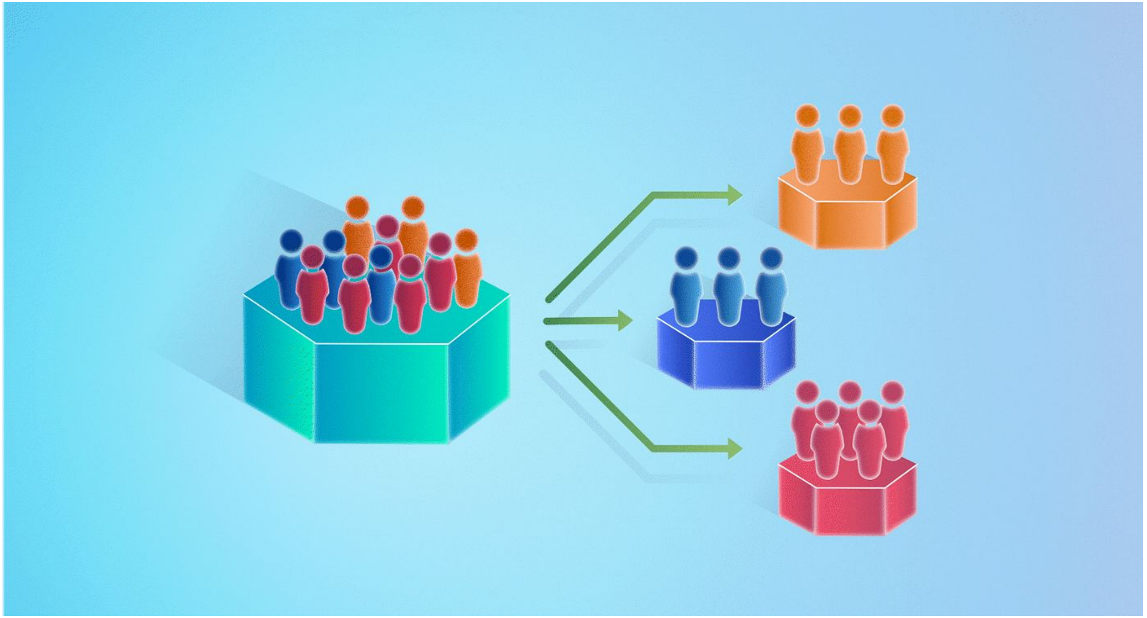
AI player ⇒ reinforcement learning





Applications and different types of learning

Customer segmentation ⇒ unsupervised learning



Source: https://medium.com/inst414-data-science-tech/customer-segmentation-bdd7e1053cb3

One more definition tentative

- “The field of study that gives computers the ability to learn without being explicitly programmed” (Arthur Samuel, 1959)

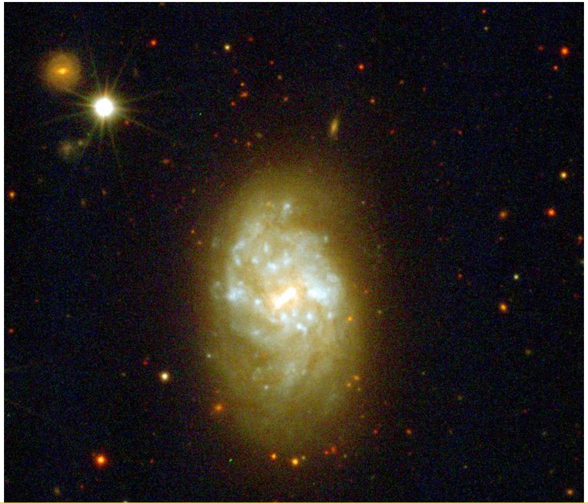
- “A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at task T, as measured by P, improves with experience E.” (Tom Mitchell, 1997)

Example:
If the computer plays many games (experience E),
and over time it wins more games (performance P improves)
at the task of playing chess (task T), then the program has learned.

Two common types of tasks

Classification

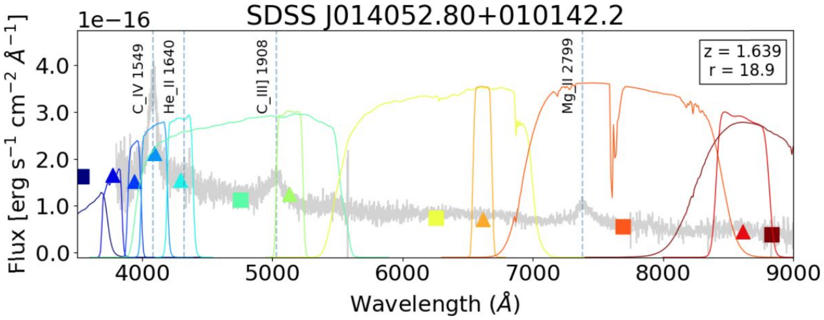
What astronomical object is this?



Y = {star, galaxy, quasar, ...}
discrete values

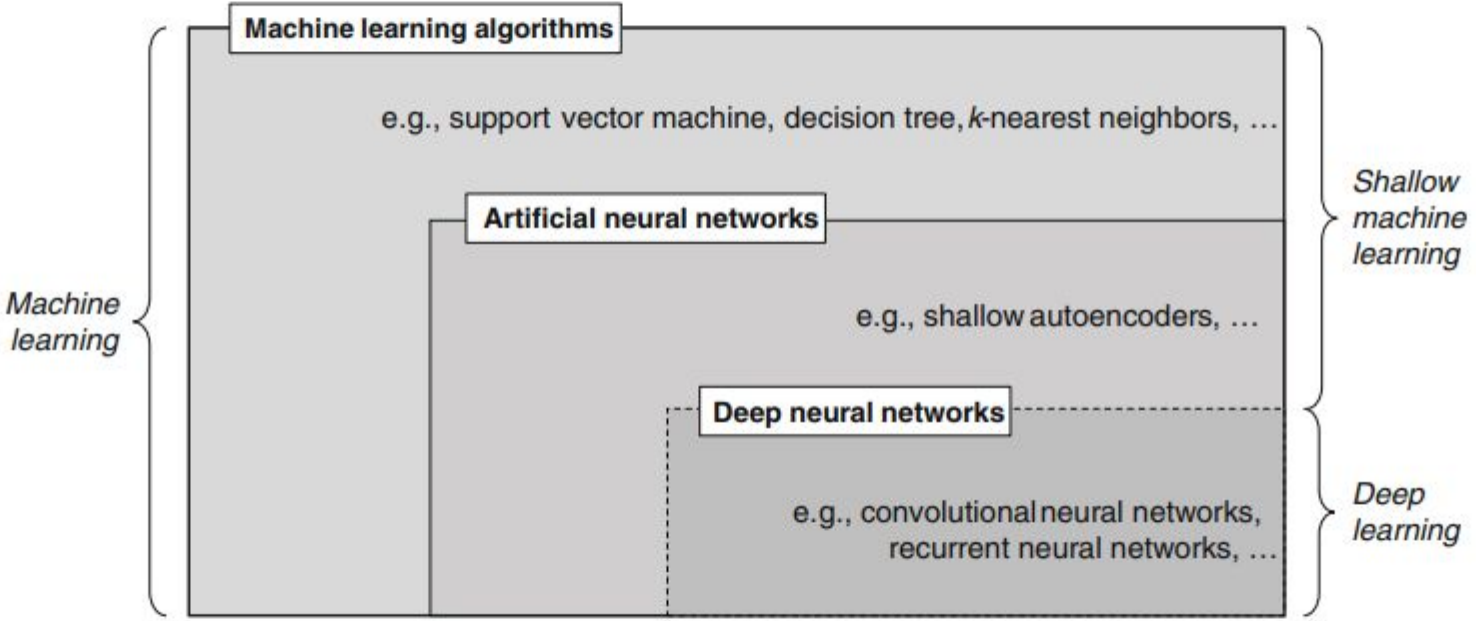
Regression

If we were only given the photometric points, can we estimate redshift (z)?



Y = $\mathbb{R}$
continuous values

Shallow Machine Learning x Deep Learning



Venn diagram of machine Machine learning algorithms learning concepts and classes (inspired by Goodfellow et al. 2016, p. 9). Source: Janiesch, Zschech & Heinrich, 2021

Supervised Learning

For this lecture, I will only consider a supervised learning scenario. Let's formally define this concept:

Consider $\mathcal{X}$ the input data space and $\mathcal{Y}$, the output space

$x^{(i)}$ is the i-th element of $\mathcal{X}$
$y^{(i)}$ is the i-th element of $\mathcal{Y}$

→ each pair ($x^{(i)}$, $y^{(i)}$) are element of the dataset D

In other words:

$$D = \{(x^{(i)}, y^{(i)}): x^{(i)} \in \mathcal{X} \text{ and } y^{(i)} \in \mathcal{Y}, i = 1,\dots, N\}$$

Supervised Learning

I will show how later in this lecture!

We want to find a function f that relates each $x^{(i)}$ to $y^{(i)}$ in the **best way possible**:

$$f : \mathcal{X} \rightarrow \mathcal{Y}$$

In "classical" statistical analysis, knowing f explicitly is important to interpret the relationship between $\mathcal{X}$ and $\mathcal{Y}$ and make a statistical inference.

In machine learning, we often lose interpretability (f cannot be written down) but increase prediction power.

Supervised Learning

x → nature → y

"Classical" statistics

x → linear regression $y = 0.034 + 7.2x$ → y

Machine Learning

x → ? → y

linear regression
4.3% accuracy in predicting y

Reading recommendation:
"Statistical modeling: the two cultures", Leo Breiman 2001
https://www2.math.uu.se/~thulin/mm/breiman.pdf

Supervised Learning

I will show how later in this lecture!

We want to find a function f that relates each $x^{(i)}$ to $y^{(i)}$ in the **best way possible**:

$$f : \mathcal{X} \rightarrow \mathcal{Y}$$

In "classical" statistical analysis, knowing f explicitly is important to interpret the relationship between $\mathcal{X}$ and $\mathcal{Y}$ and make a statistical inference.

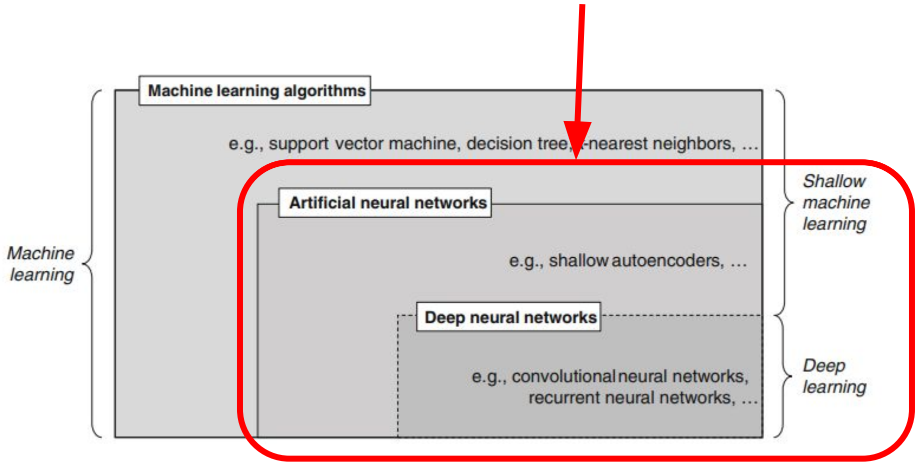
In machine learning, we often lose interpretability (f cannot be written down) but increase prediction power.

$\Rightarrow f$ has to map well not only $x^{(i)}$ and $y^{(i)}$ but also "unseen" x and y

Supervised Learning

And how to find f (not necessarily explicitly)? Answer: using learning algorithms

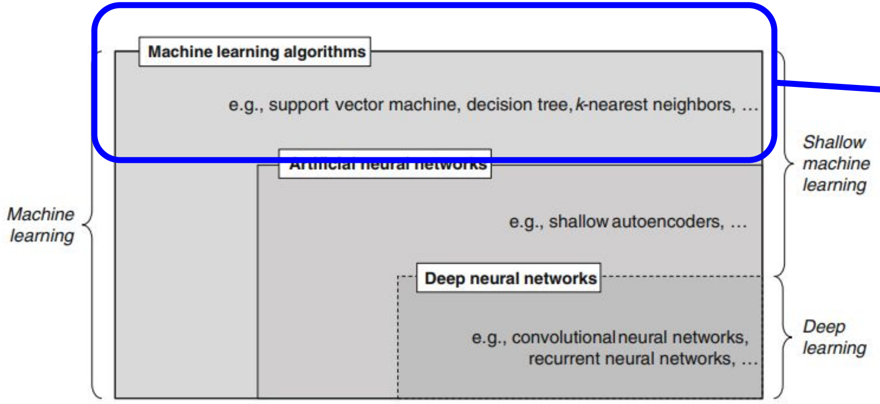
There are many options of algorithms. Some will be tackled with more detail during this course with more focus to



Supervised Learning

And how to find f (not necessarily explicitly)? Answer: using learning algorithms

There are many options of algorithms. Some will be tackled with more detail during this course with more focus to

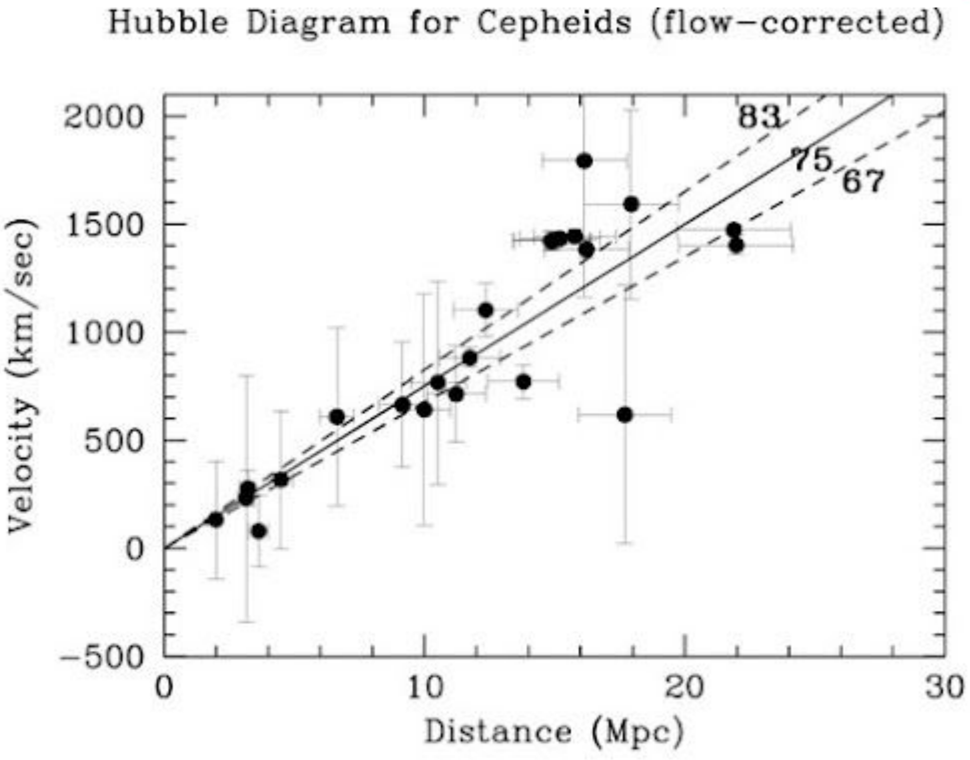


Also very useful in astronomical science! But as we do not have much time to go through most of them, I will only show two very simple models: **Linear Regression** and **Logistic Regression**

(Note: in real life, these two do not usually provide good predictions but they are key to understand neural networks. Also, the concepts that I will show will serve to any other algorithms, with proper modifications.)

Linear Regression

Note: this is an example of a classical statistical analysis using linear regression that gives the relationship $v = H_0 d$. Accurate predictions were not the goal!



Linear Regression

Consider a two-dimensional case, i.e., $(x^{(i)}, y^{(i)}) \in \mathbb{R}^2$, n = 1, 2, ... , N.

A family of hypothetical functions $h : \mathbb{R} \rightarrow \mathbb{R}$ can be written as:

$$h_{\mathbf{w}}(x) = w_0 + w_1 x$$

How can we find w_0 and w_1?

Answer: by minimizing a loss function $\mathcal{L}$

$$\mathcal{L}(w_0, w_1) = \sum_{i=1}^{N} (w_0 + w_1 x^{(i)} - y^{(i)})^2$$

Linear Regression: Analytical Solution (OLS)

Minimizing a loss function $\mathcal{L}$ means taking its derivative:

$$\hat{w} = \arg\min_w \mathcal{L}(w_0, w_1)$$

For a 2-dimensional case:

$$\hat{w_0} = \bar{Y} - w_1 \bar{X}$$

$$\hat{w_1} = \frac{\sum_{i=1}^{N} (x^{(i)} - \bar{X})(y^{(i)} - \bar{Y})}{\sum_{i=1}^{N} (x^{(i)} - \bar{X})^2}$$

Analytical solution for any size of w (Ordinary Least Squares estimator):

$$\hat{w} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

There are some advantages of having this analytical solution, but I won't dive into these details here!

Numerical solution (Gradient descent)

Why do we care about a numerical solution for linear regression if we have an analytical one?

$$\hat{w} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

inverting X^TX can be very computationally expensive! OLS works well for small datasets

Plus, OLS only works for linear models. If you have very large dataset, many features and wants to fit a nonlinear model, you need a numerical solution method.

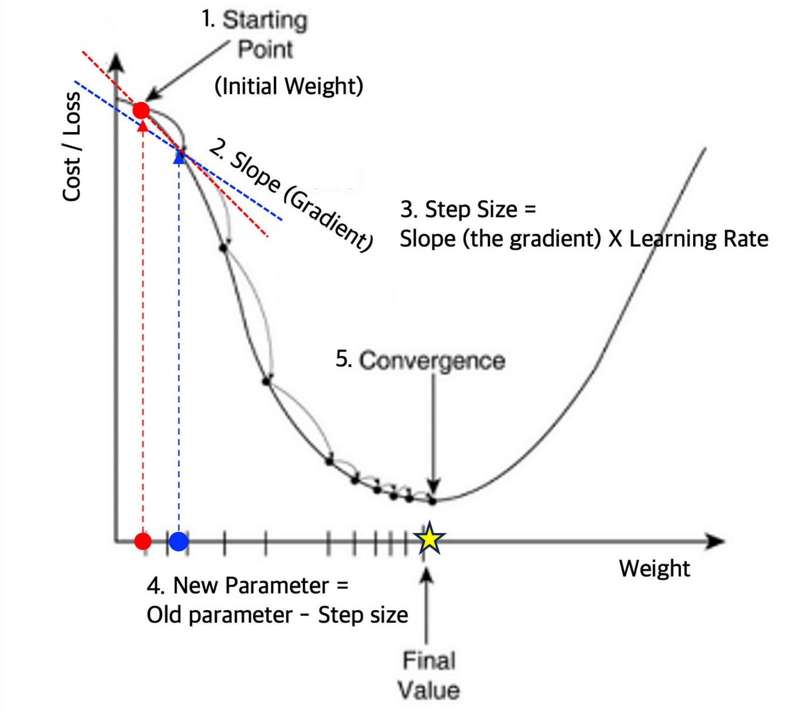
This will be very important for neural networks!

Visualization of Gradient Descent (1-dimensional)

Imagine that the curve in this illustration is the loss function

$$\mathcal{L}(w_0, w_1) = \sum_{i=1}^{N} (w_0 + w_1 x^{(i)} - y^{(i)})^2$$

The idea of applying a numerical solution method is to find the minimum by iterative calculations



source: https://medium.com/@msong507/gradient-descent-intuitive-overview-43e7869e1545

Numerical solution (Gradient descent)

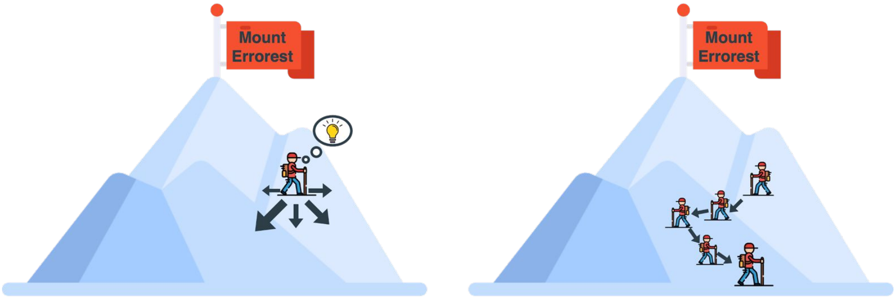
Iterative algorithm:

- Initialize parameters **w**(0) with randoms
- Compute the gradient of $\mathcal{L}$, i.e. $\nabla\mathcal{L}$ (**w**(0))
- Update the parameters **w** as follows:

$$w(1) = w(0) - \eta \, \nabla\mathcal{L}\,(w(0))$$

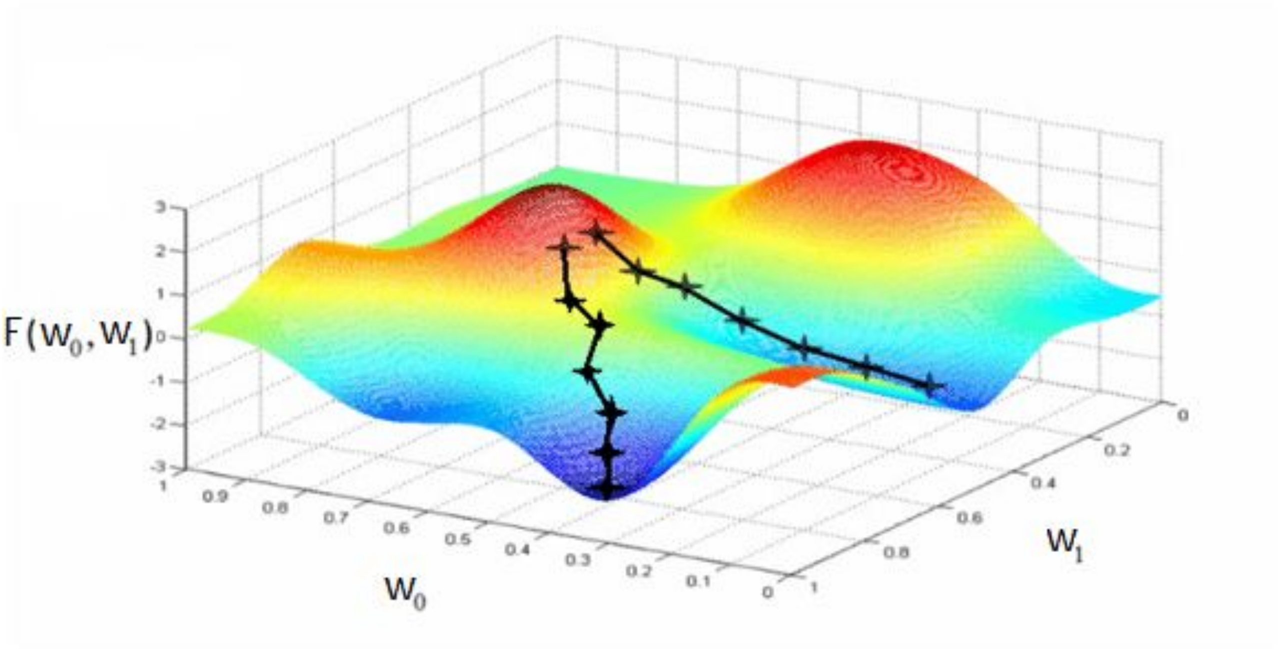
learning rate: arbitrary value

- Repeat until convergence



Source: https://livebook.manning.com/book/grokking-machine-learning/appendix-b/v-14/13

Visualization of Gradient Descent (2-dimensional)



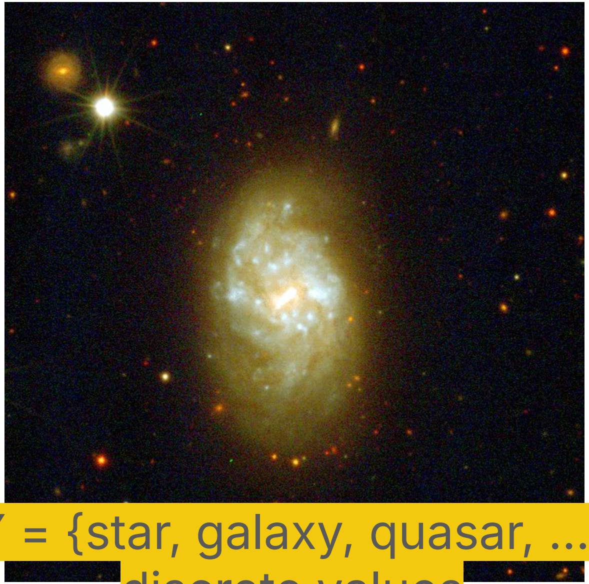
Source:
https://www.researchgate.net/publication/322203555_Memristive_crossbar_arrays_for_machine_learning_systems

Two common types of tasks

We talked about this case!

Classification

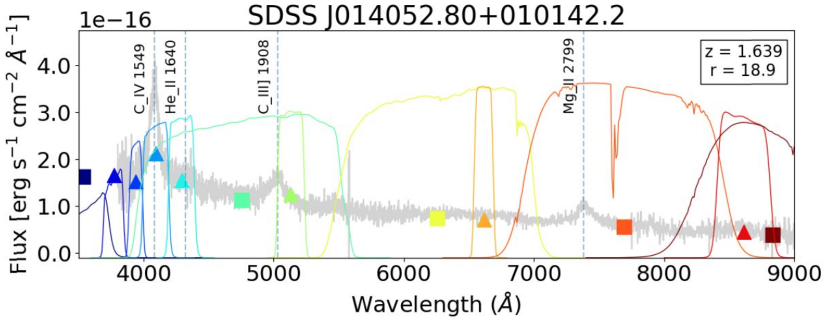
What astronomical object is this?



Y = {star, galaxy, quasar, ...}
discrete values

Regression

If we were only given the photometric points, can we estimate redshift (z)?



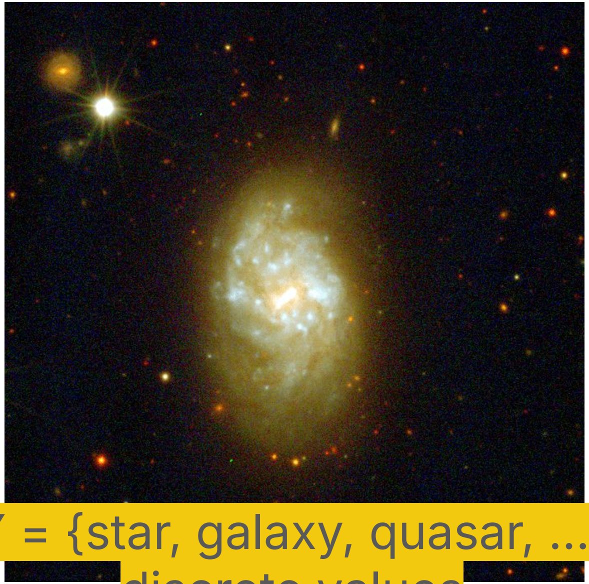
$Y = \mathbb{R}$
continuous values

Two common types of tasks

Now let's talk about this one

Classification

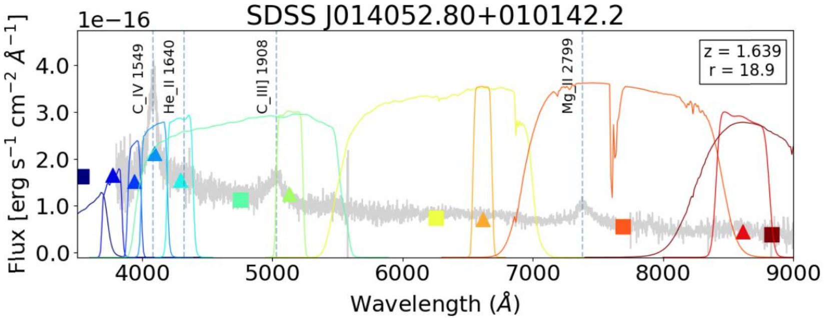
What astronomical object is this?



Y = {star, galaxy, quasar, ...}
discrete values

Regression

If we were only given the photometric points, can we estimate redshift (z)?



Y = $\mathbb{R}$
continuous values

Logistic Regression

Note: Here, "regression" does not refer to a regression task as we defined in the previous slides. In Statistics, regression is a statistical process to relate a dependent variable to one or more independent variables, regardless if y is continuous or not.

We just saw that learning algorithms are purely based on linear algebra and differential calculus!

How can we tackle a classification task if our output are now words?

Answer: use numbers instead of words, e.g. {0, 1}
(same idea goes to natural language processing, e.g. chatgpt!)

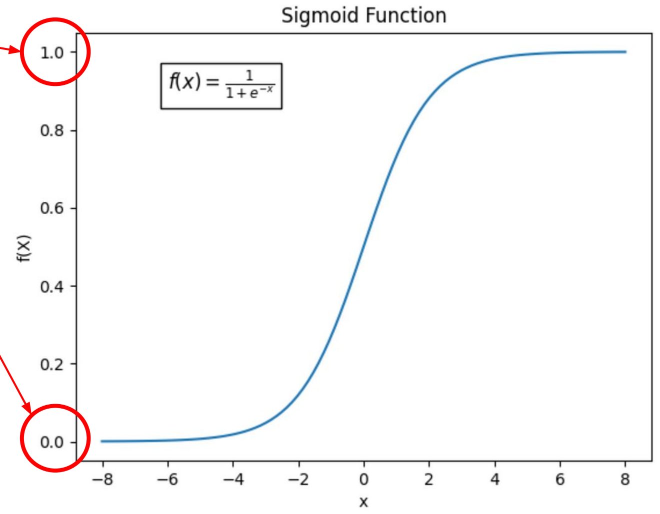
Logistic regression only works for binary classification. We want to predict:

$$P(y = 1|x) \begin{cases} \text{class 1 if probability >= 50\%} \\ \text{class 0 if probability < 50\%} \end{cases}$$

Logistic Regression

As $0 < P(y|x) < 1$ (from Probability Theory), we use the logistic (or sigmoid) function to keep the output between 0 and 1:

$$\sigma(z) = \frac{1}{1+e^{-z}}$$



and $f: \mathcal{X} \rightarrow \mathcal{Y}$ will be approximated by

$$h_{\mathbf{w}}(\mathbf{x}) = \theta(\mathbf{w}^T\mathbf{x})$$

Logistic Regression

Logistic Regression can be extended to K classes ⇒ Multinomial Logistic Regression

In this case, we use the **softmax** function instead of the sigmoid.

$$\sigma(z_k) = \frac{e^{z_k}}{\sum_{j=1}^{K} e^{z_k}}$$

Logistic Regression

Without going in detail on how we get this expression, this is the cross-entropy loss:

$$\mathcal{L}(\mathbf{w}) = -\frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K} y_k^{(n)} \, log \, \hat{p}_k^{(n)}$$

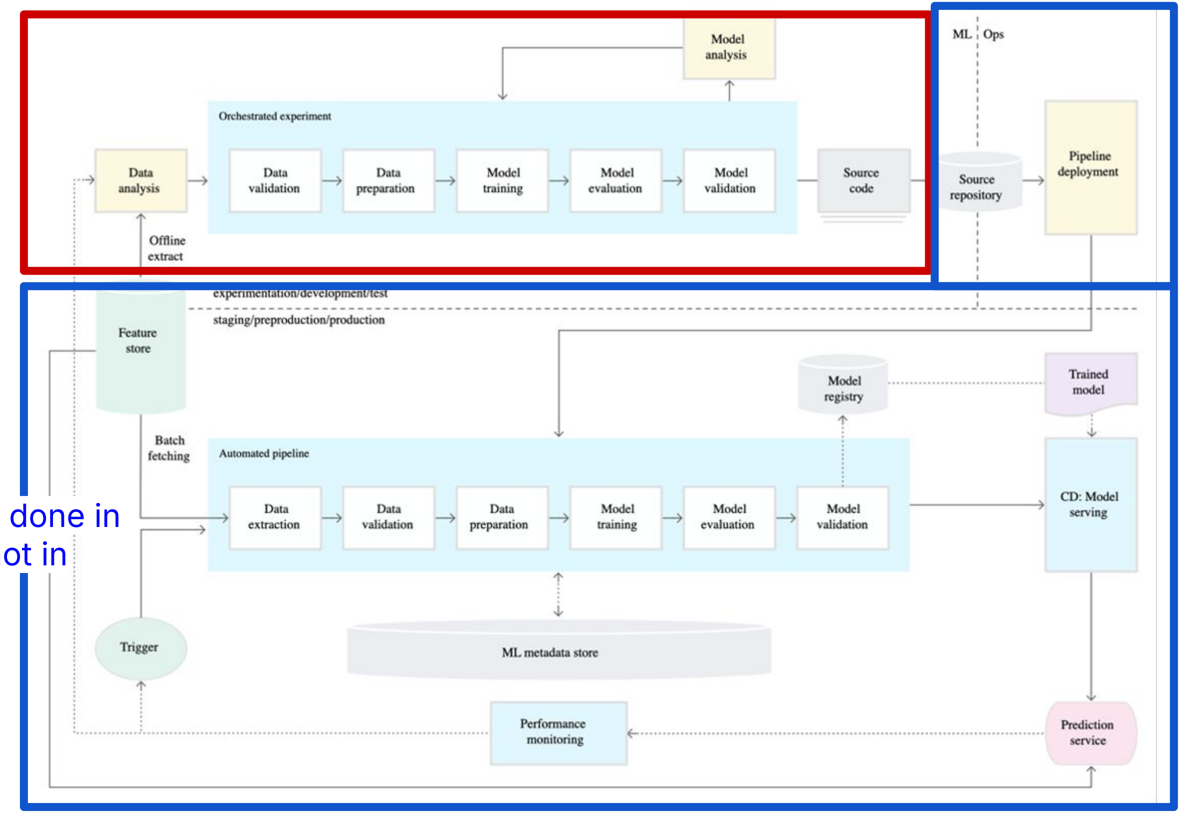
where

$$\hat{p}_k = \hat{P}(y = k \mid \mathbf{x}) = \frac{e^{\mathbf{w}_k^T \mathbf{x}}}{\sum_{j=1}^{K} e^{\mathbf{w}_k^T \mathbf{x}}}$$

The cross-entropy loss can be minimized with gradient descent.

A Machine Learning project infrastructure

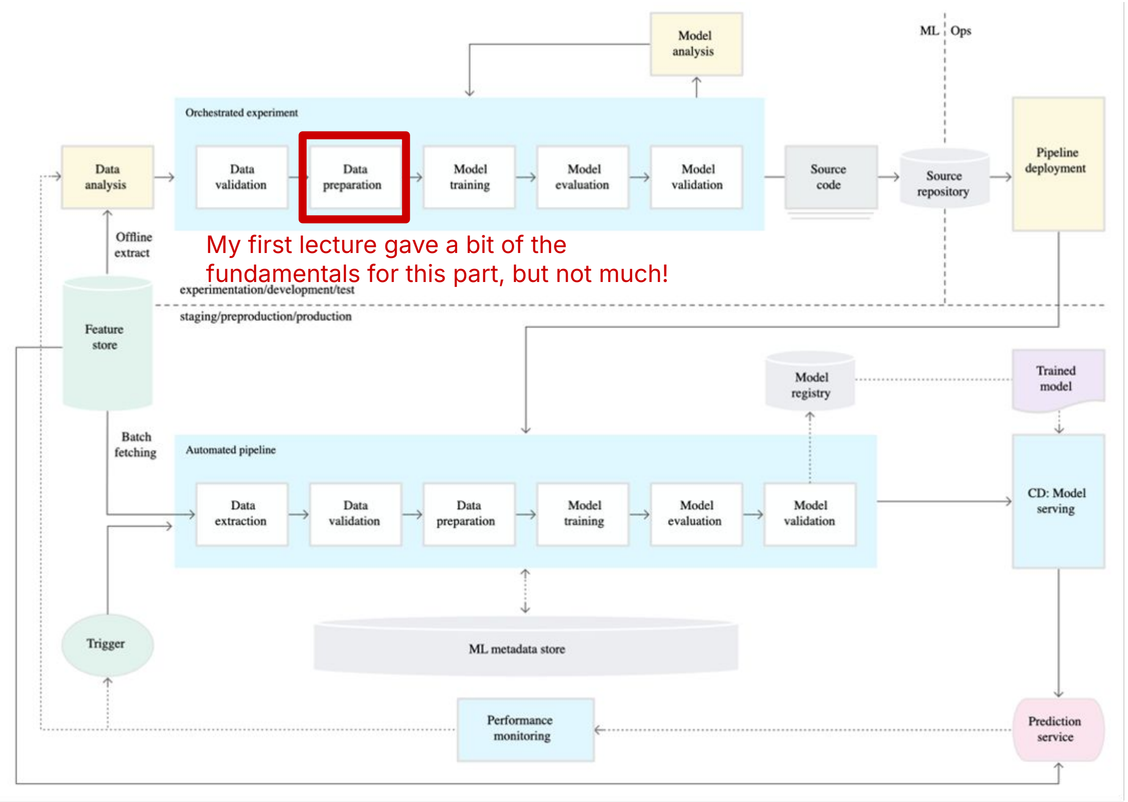
this part is what we commonly do in scientific applications



this part is commonly done in tech companies but not in Astronomy... yet.

Source: https://insights.sei.cmu.edu/blog/a-hitchhikers-guide-to-ml-training-infrastructure/

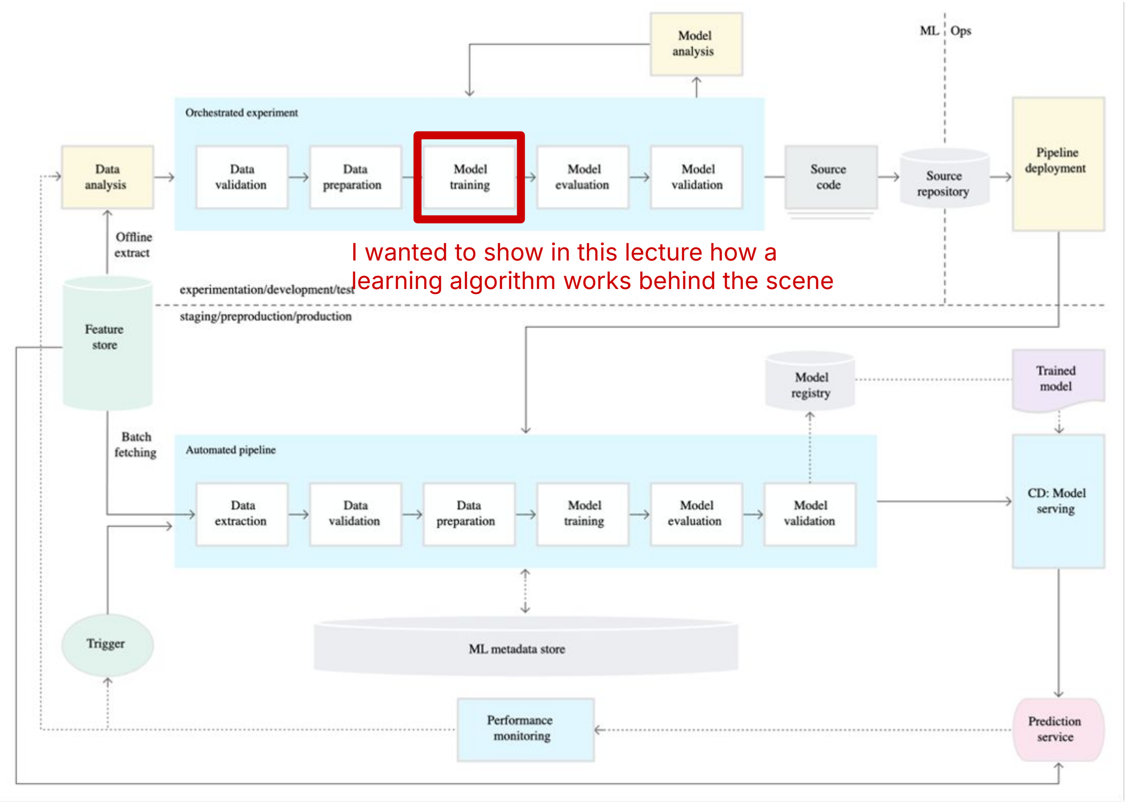
A Machine Learning project infrastructure



My first lecture gave a bit of the fundamentals for this part, but not much!

Source: https://insights.sei.cmu.edu/blog/a-hitchhikers-guide-to-ml-training-infrastructure/

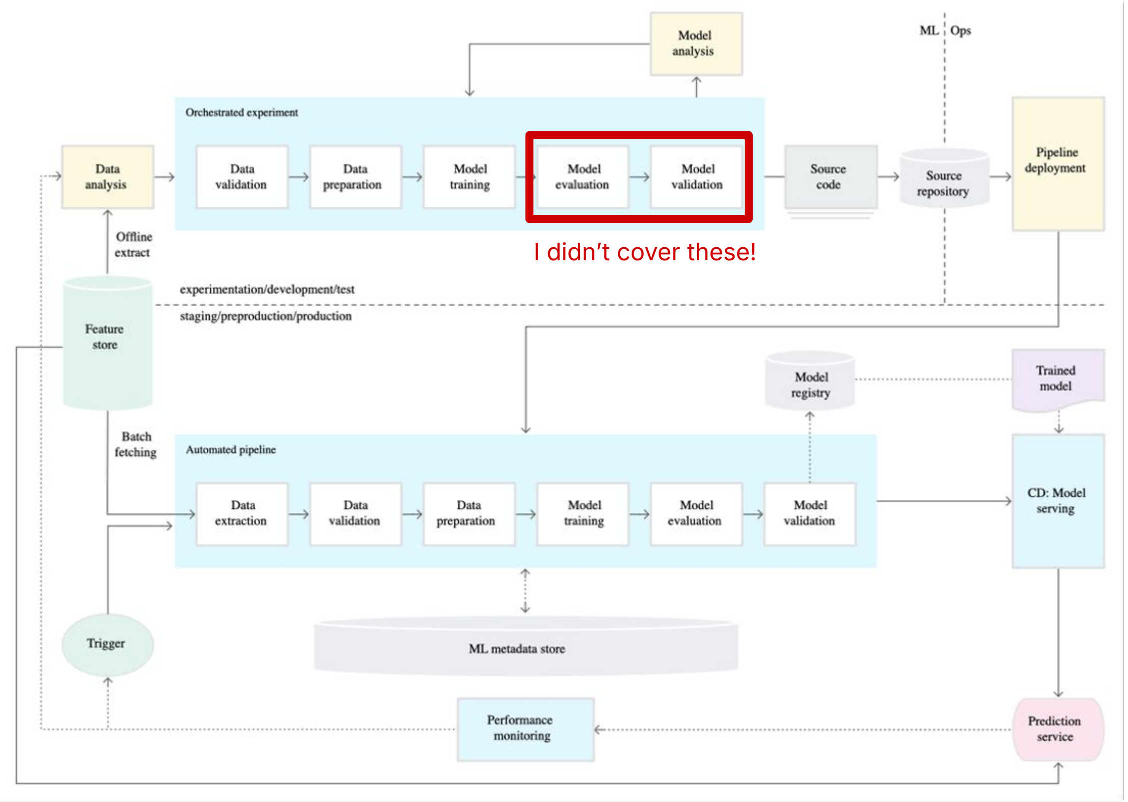
A Machine Learning project infrastructure



I wanted to show in this lecture how a learning algorithm works behind the scene

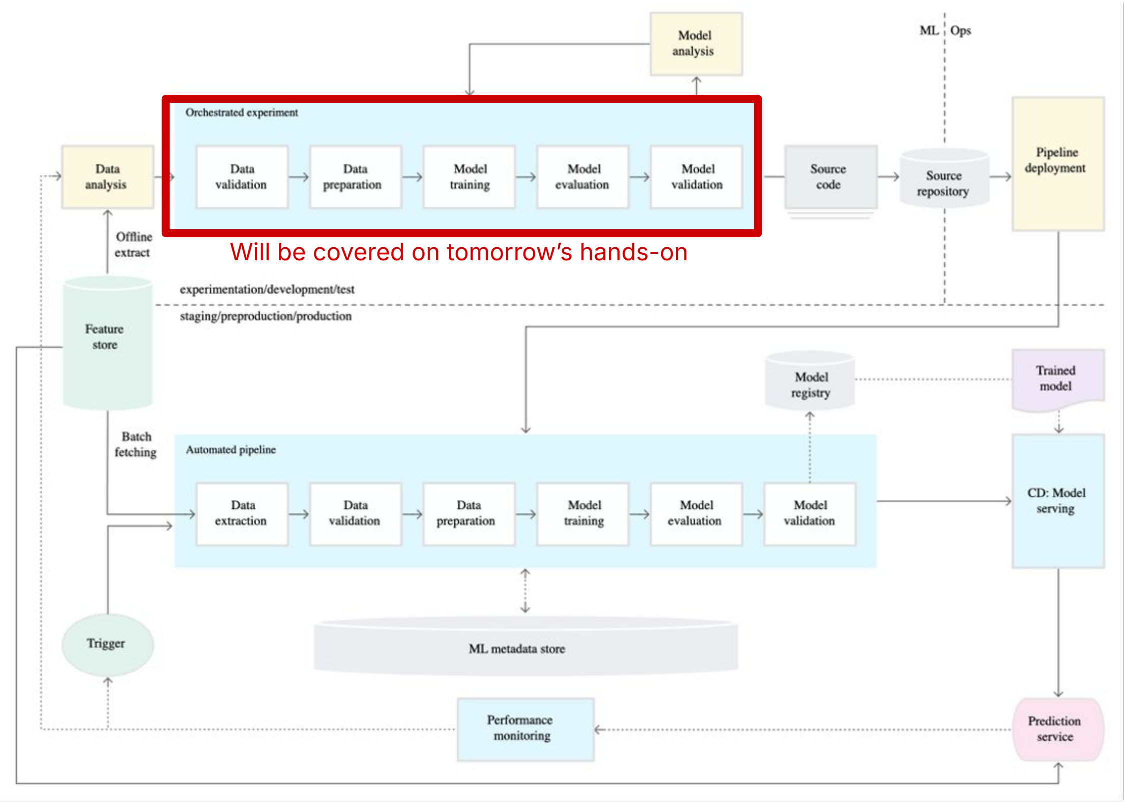
Source: https://insights.sei.cmu.edu/blog/a-hitchhikers-guide-to-ml-training-infrastructure/

A Machine Learning project infrastructure



Source: https://insights.sei.cmu.edu/blog/a-hitchhikers-guide-to-ml-training-infrastructure/

A Machine Learning project infrastructure



Source: https://insights.sei.cmu.edu/blog/a-hitchhikers-guide-to-ml-training-infrastructure/

Machine Learning Glossary

Terms and concepts: https://developers.google.com/machine-learning/glossary

Mathematical Notation: https://nthu-datalab.github.io/ml/slides/Notation.pdf

Compilation of cheatsheets:
https://github.com/FavioVazquez/ds-cheatsheets?tab=readme-ov-file

Lecturers may use very different mathematical notations from each other. So if you feel lost, don't be afraid of asking "what does [insert a weird symbol here] mean?"

S-PLUS

Contacts

Email: lilianne.nakazono@gmail.com

GitHub: https://github.com/marixko

Website: https://marixko.github.io