Lecture #1

Welcome to COGS 118A! An introduction to Supervised Machine Learning Algorithms

Jason G. Fleischer, Ph.D.

Asst. Teaching Professor
Department of Cognitive Science, UC San Diego

jfleischer@ucsd.edu

@jasongfleischer

https://jgfleischer.com

Some slides in this presentation are from
Zhuowen Tu, Sebastian Rashkha, or Brad Voytek

This class

syllabus link here

github link here

Time to scroll the syllabus and Canvas

How to succeed in this course

- Watch the videos before lecture
- Explore beyond the basic material
  - Readings and other extras in this course
  - Keep going!
- Play
  - with the Jupyter notebooks here
  - try things on your own with new data
- Participate in Campuswire and discussion sections
- Try to figure things out for a while yourself, then ask for help

Machine learning

“A breakthrough in machine learning would be worth ten Microsofts”

— Bill Gates, Microsoft Co-Founder

"Machine learning is the field of study that gives computers the ability to learn without being explicitly programmed"

— Arthur L. Samuel, AI pioneer, 1959

(This is likely not an original quote but a paraphrased version of Samuel's sentence "Programming computers to learn from experience should eventually eliminate the need for much of this detailed programming effort.")

Arthur L Samuel. "Some studies in machine learning using the game of checkers". In: *IBM Journal of research and development* 3.3 (1959), pp. 210–229.

"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 tasks in T, as measured by P, improves with experience E."

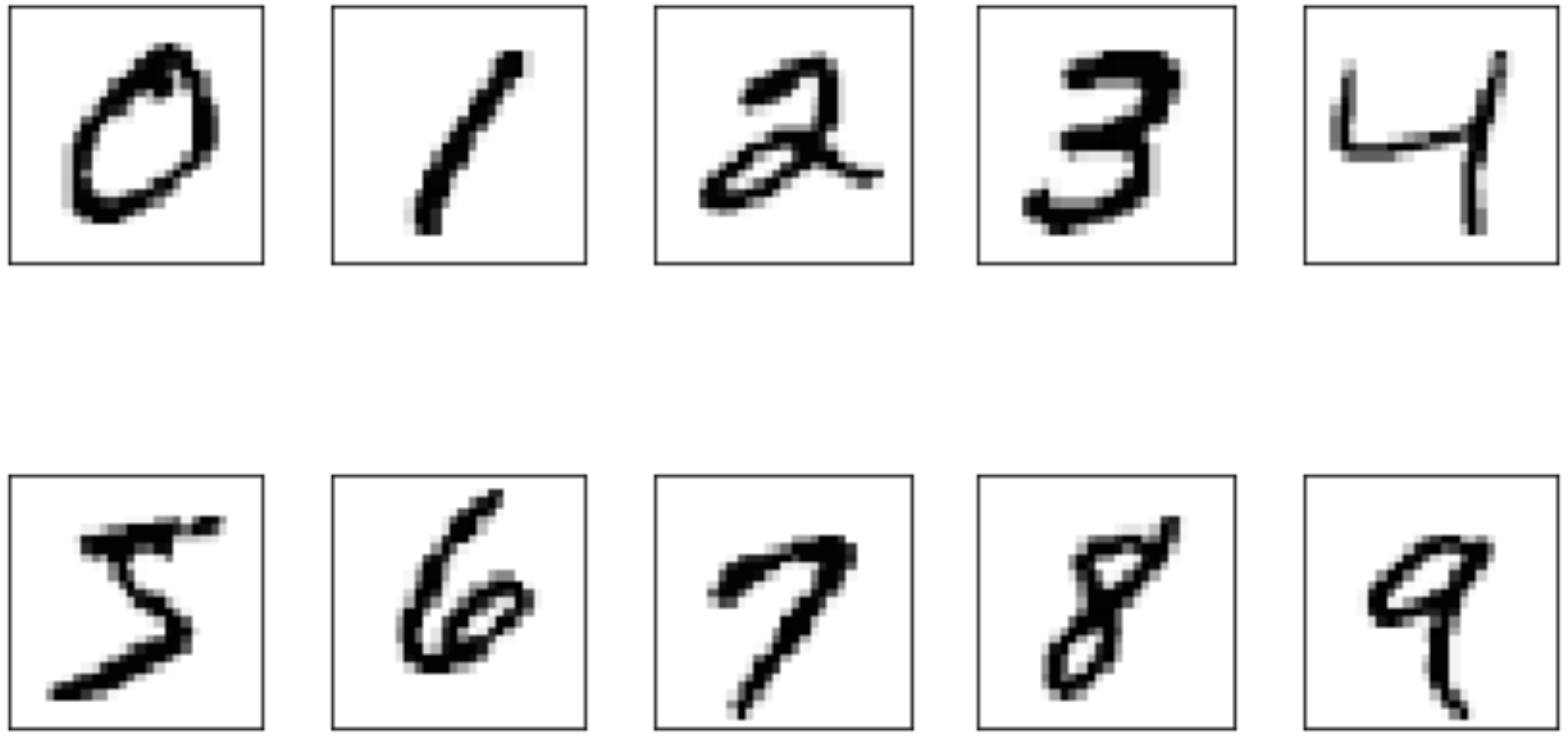
— Tom Mitchell, Professor at Carnegie Mellon University

Tom M Mitchell et al. "Machine learning. 1997". In: *Burr Ridge, IL: McGraw Hill* 45.37 (1997), pp. 870–877.

"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 tasks in T, as measured by P, improves with experience E."

— Tom Mitchell, Professor at Carnegie Mellon University

Handwriting Recognition Example:



- Task T: ?
- Performance measure P : ?
- Training experience E: ?

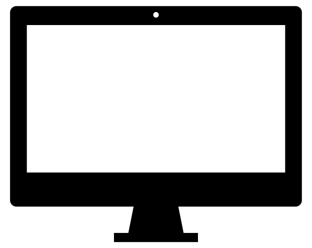
Applications of Machine Learning

Your ideas…

Applications of Machine Learning

Your ideas…

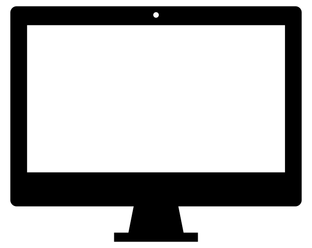
Categories of ML



Supervised

- Labelled data
- Direct feedback
- Predict, classify, or fit a model

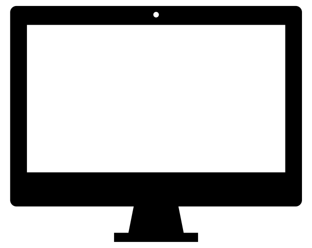
COGS 118A



Unsupervised

- No labels
- No feedback
- Find hidden structure using a model

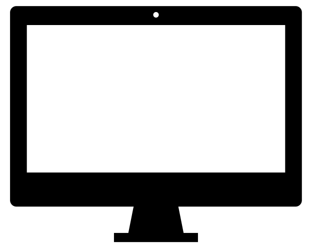
COGS 118B



Reinforcement learning

- Feedback via reward (only label)
- Learns the series of actions that lead to reward

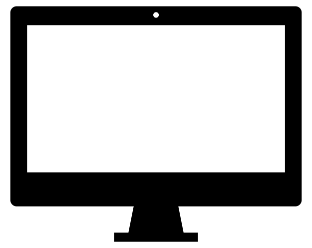
COGS 182



Neural Networks

- Mostly supervised and self-supervised. Some unsupervised
- Learn to predict/classify

COGS 181



Genetic Algorithms

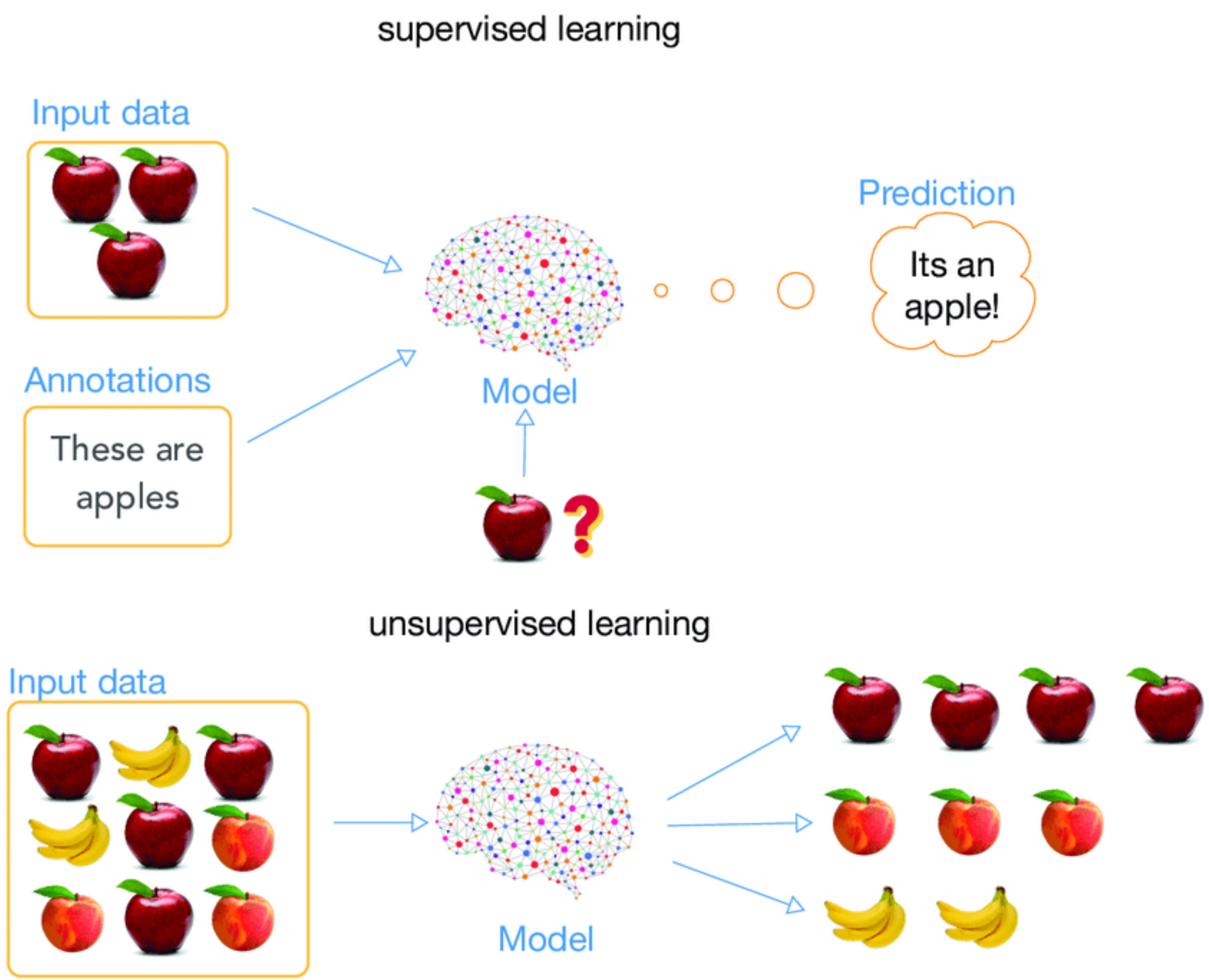
- Search for solutions through simulated natural selection
- Learn to predict/classify

COGS 186

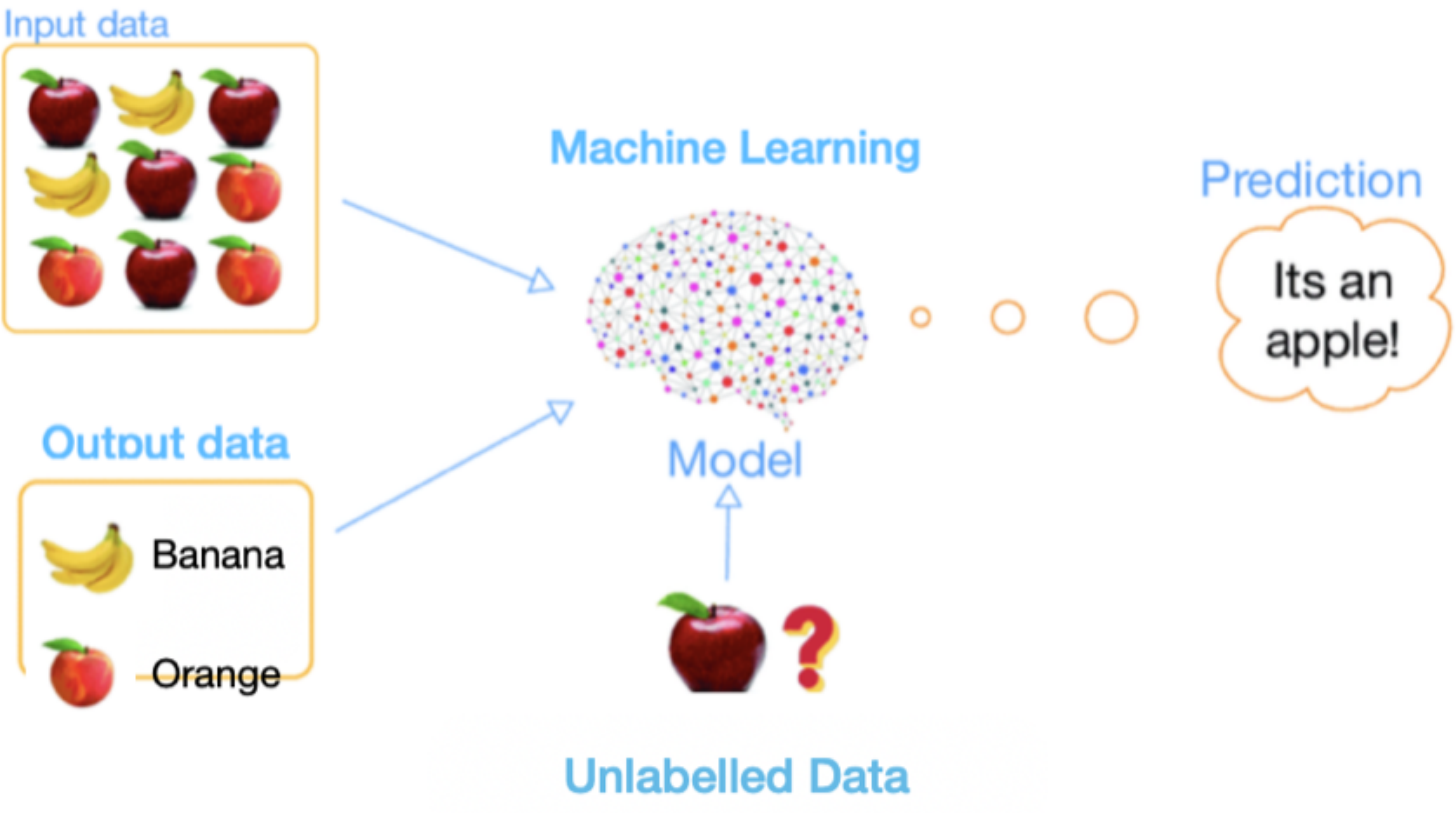
A false dichotomy? supervised vs unsupervised

COGS 118A

COGS 118B

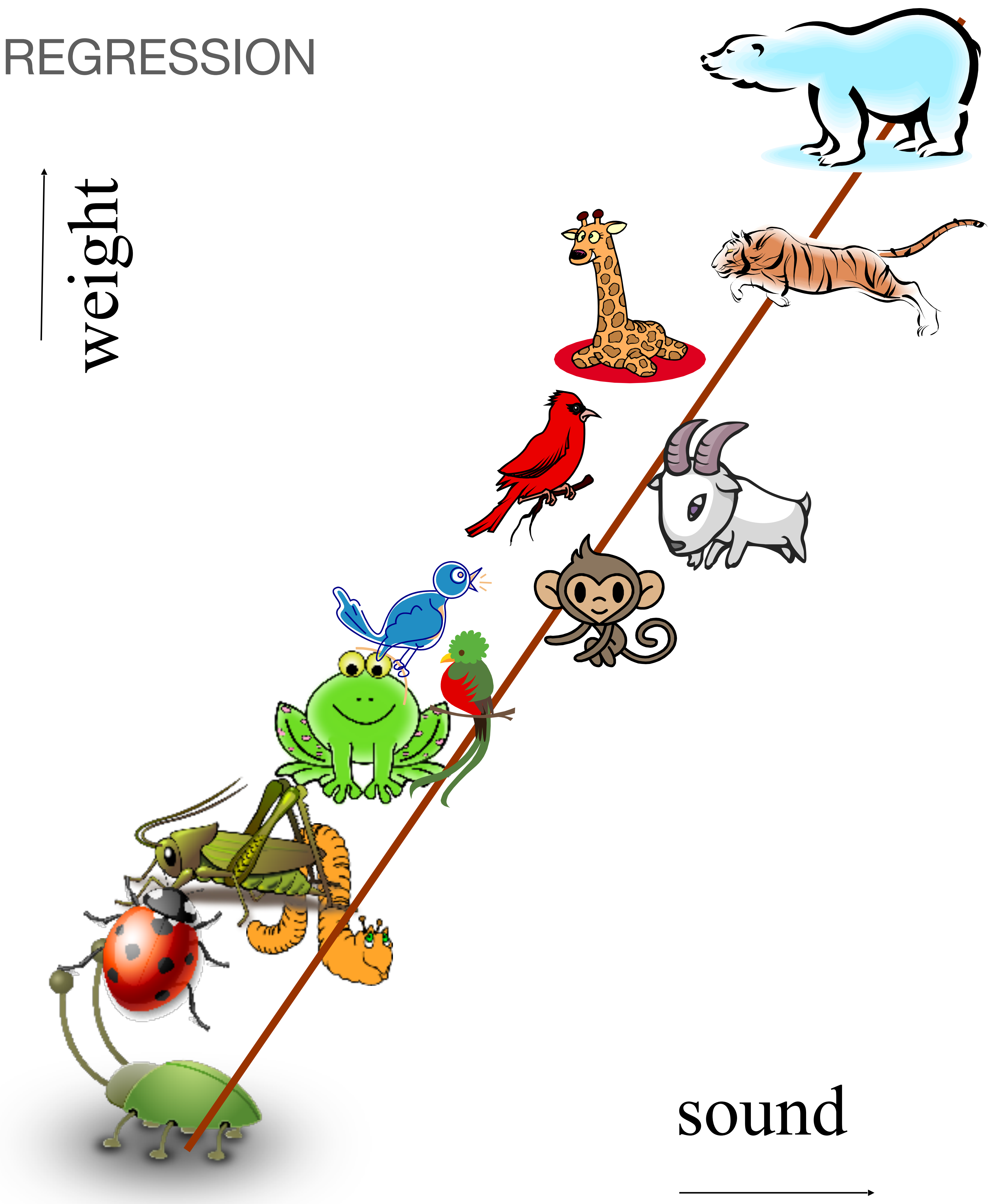


A false dichotomy? Self-supervised

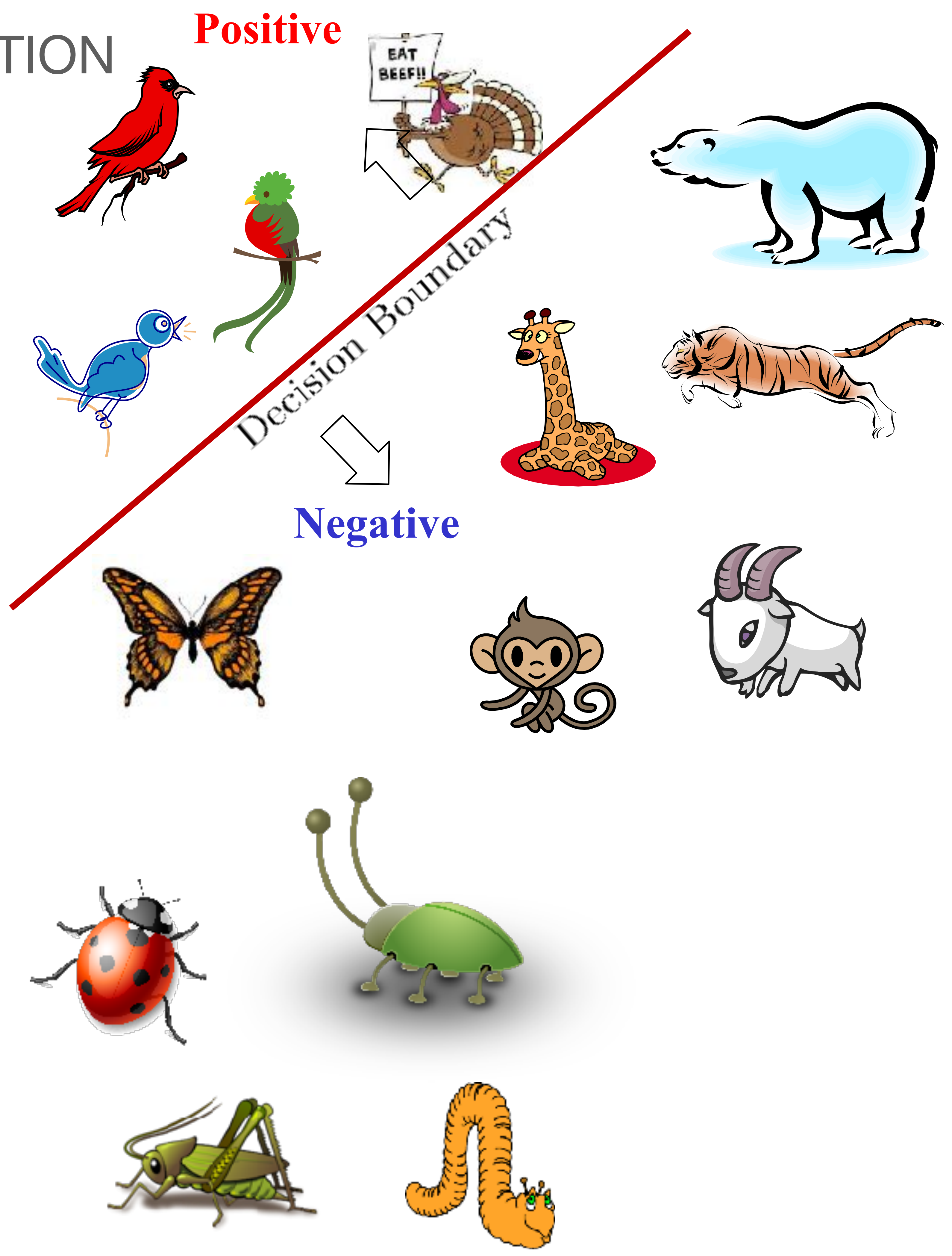


Another false dichotomy?

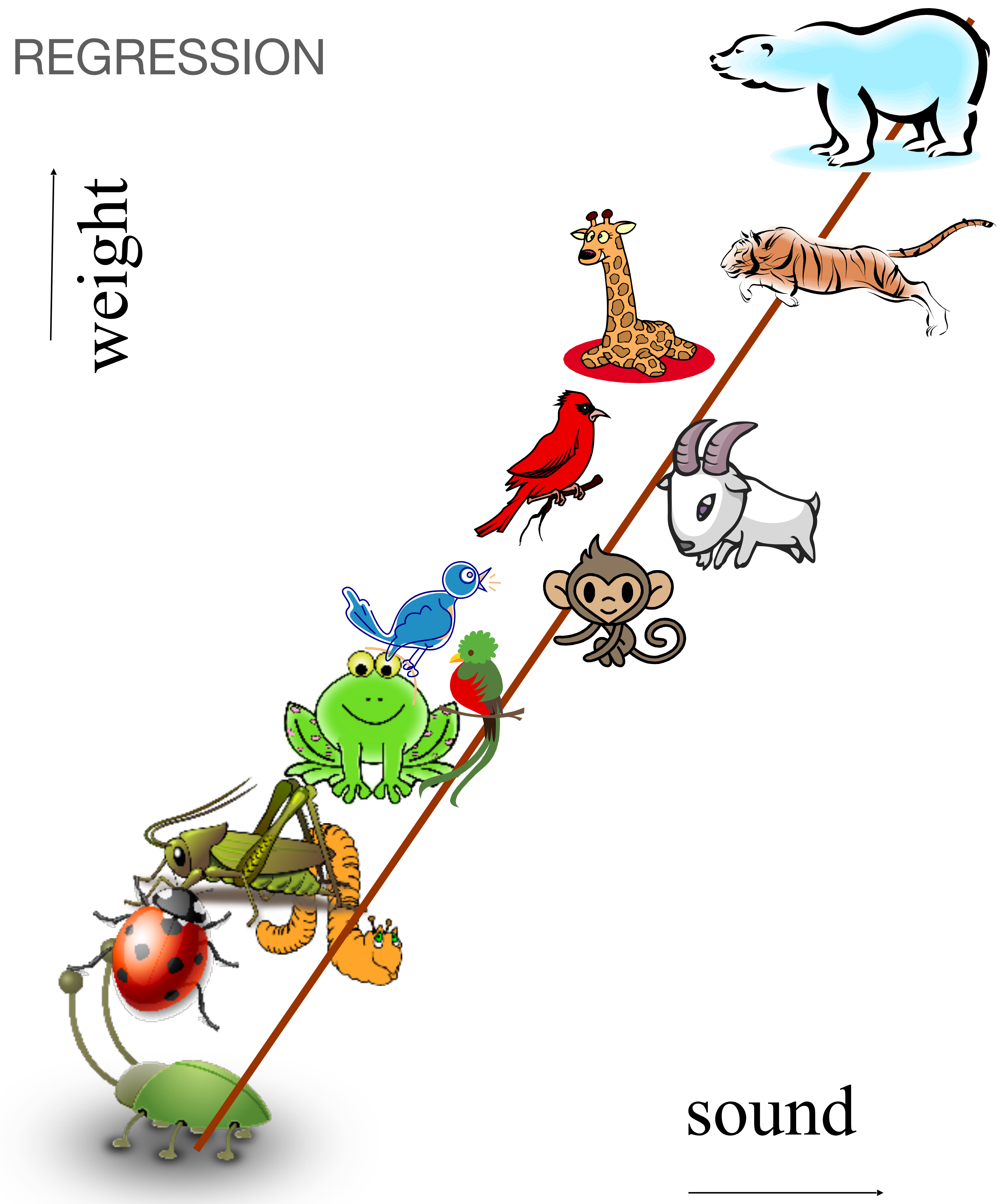
REGRESSION



CLASSIFICATION



Regression

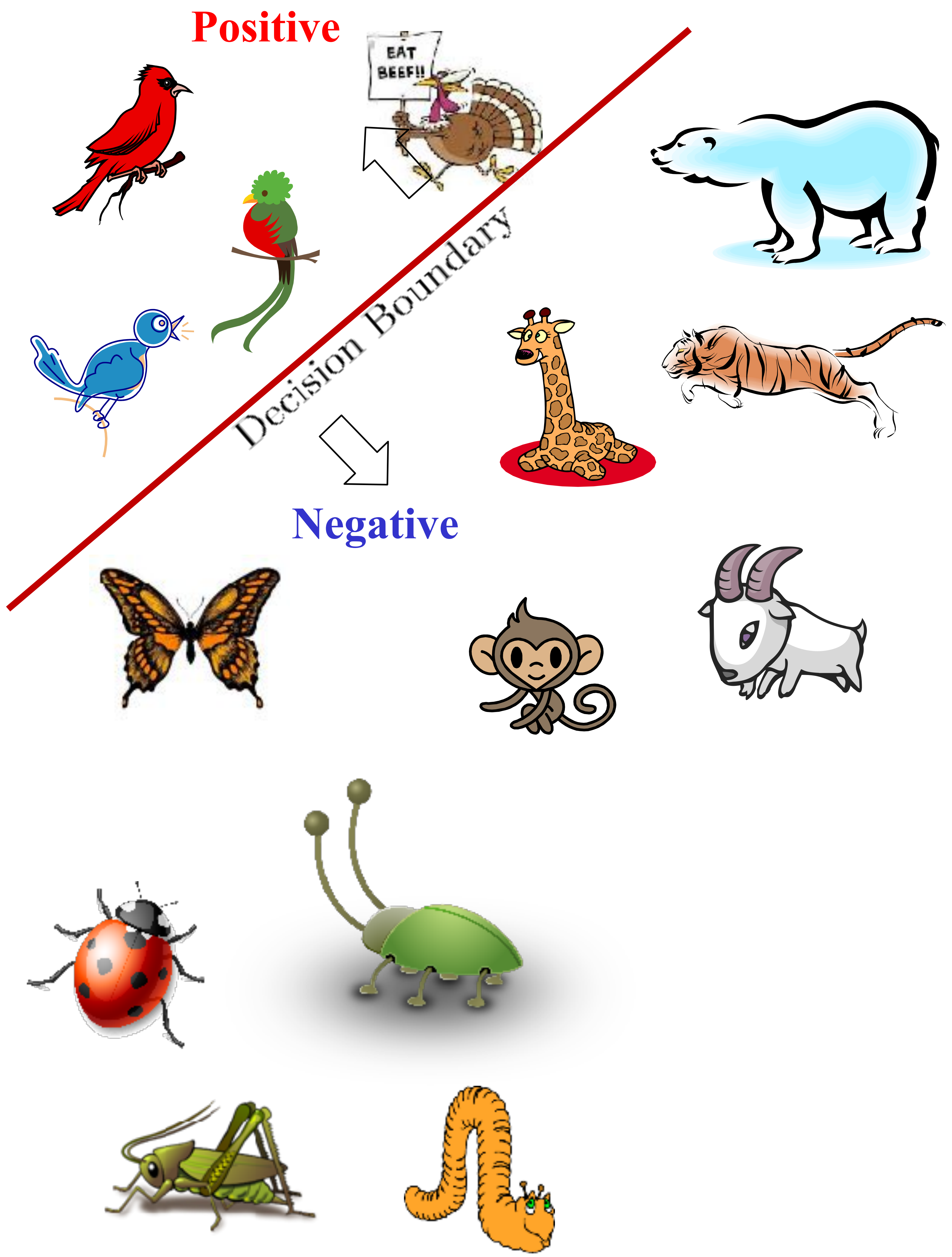


In regression, the "ground-truth" labels are continuous values that are ORDINAL

That is, you can compare $y_1 = -1$ and $y_2 = 1$ to state for example $y_1 < y_2$ $(-1 < 1)$.

$$y \in \mathbb{R}$$

Classification



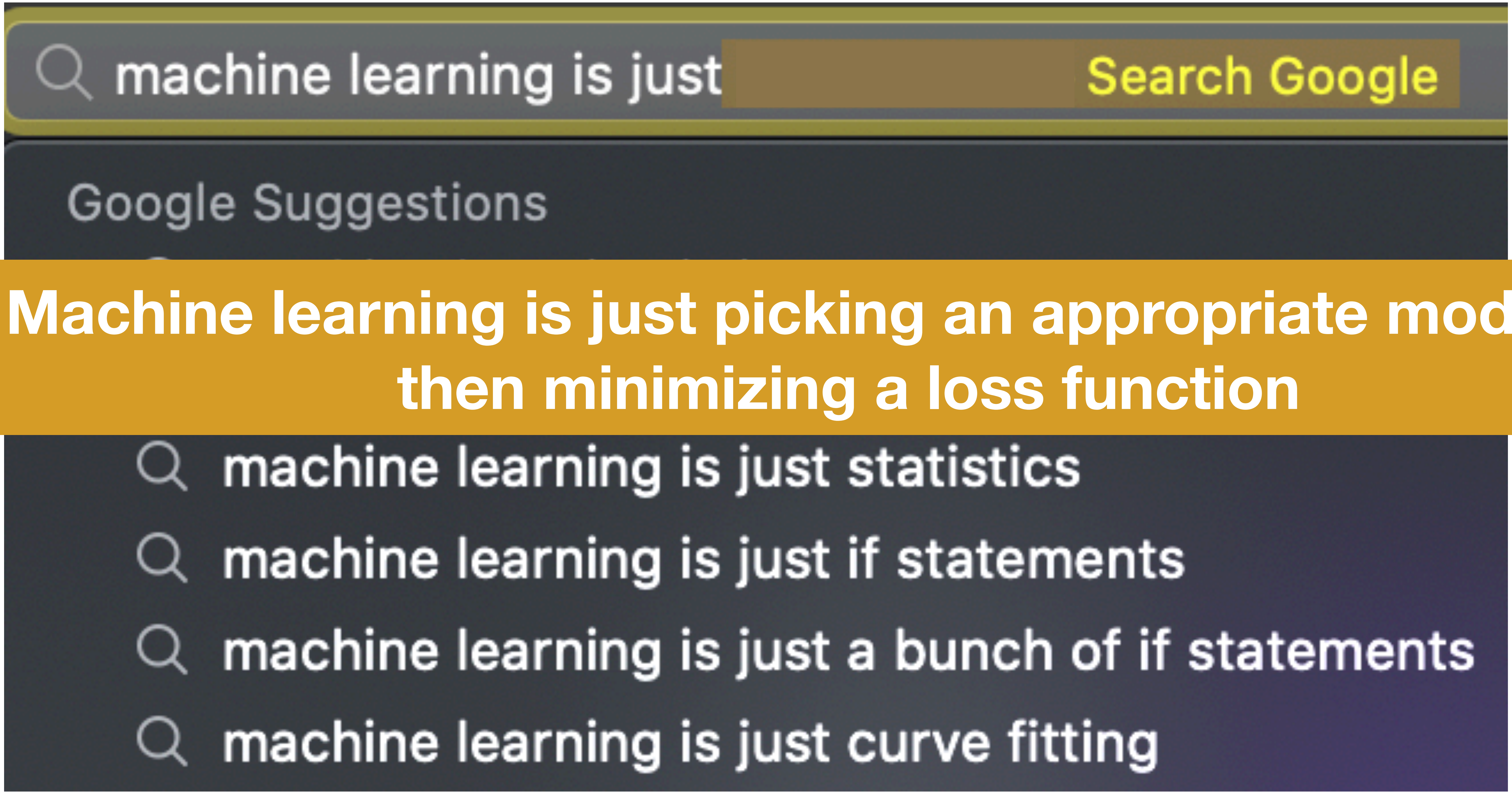
In classification, the "ground-truth" labels are categorical labels that are NOT ORDINAL

That is, you can not compare $y_1 = $ "$bird''$ and $y_2 = $ "$non - bird''$) to state for example:

"bird" < "non-bird" (even we use $+1$ and -1 to denote their labels respectively).

$$y \in \{-1, +1\}$$

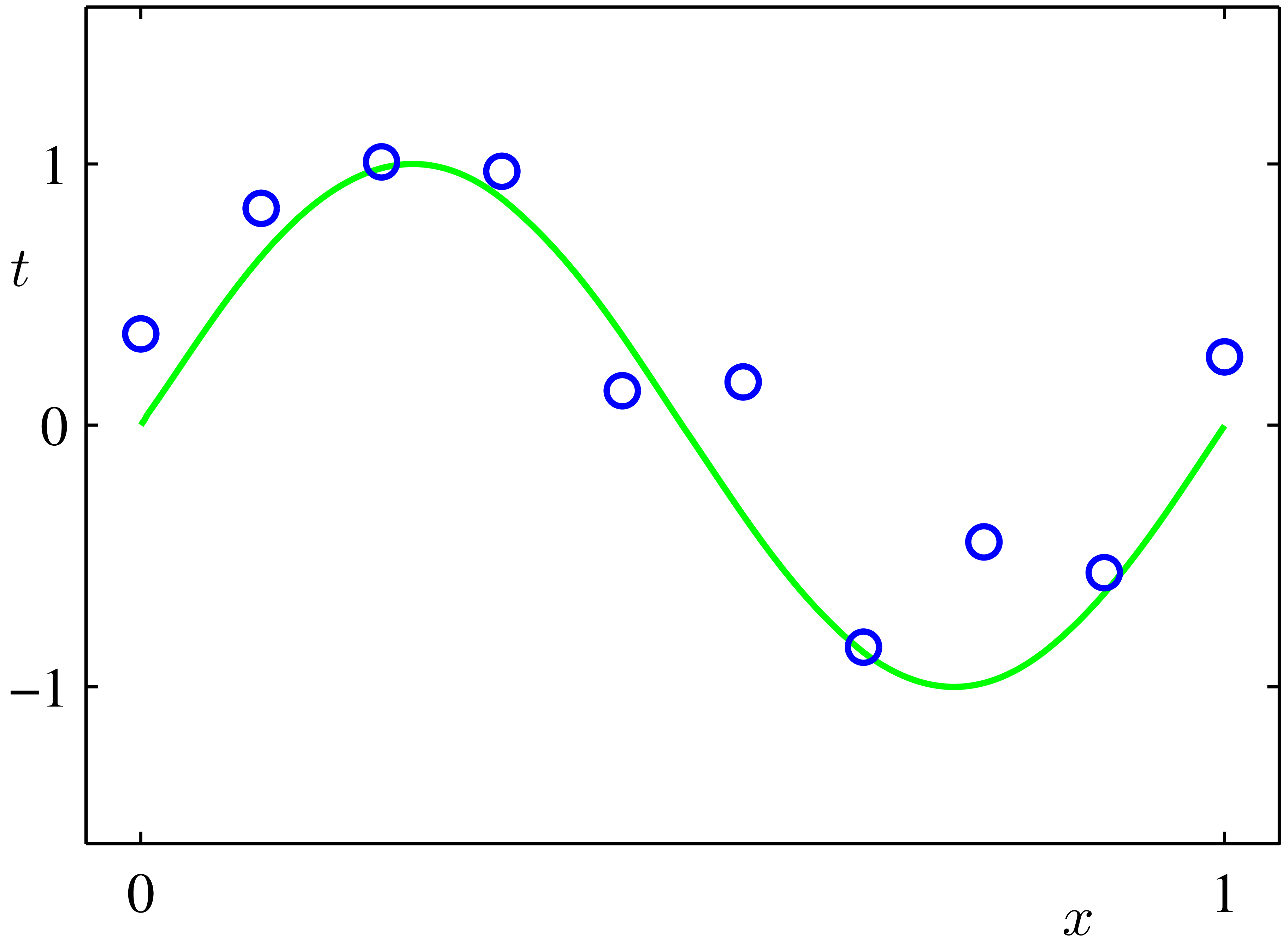
Machine learning is just picking an appropriate model and then minimizing a loss function

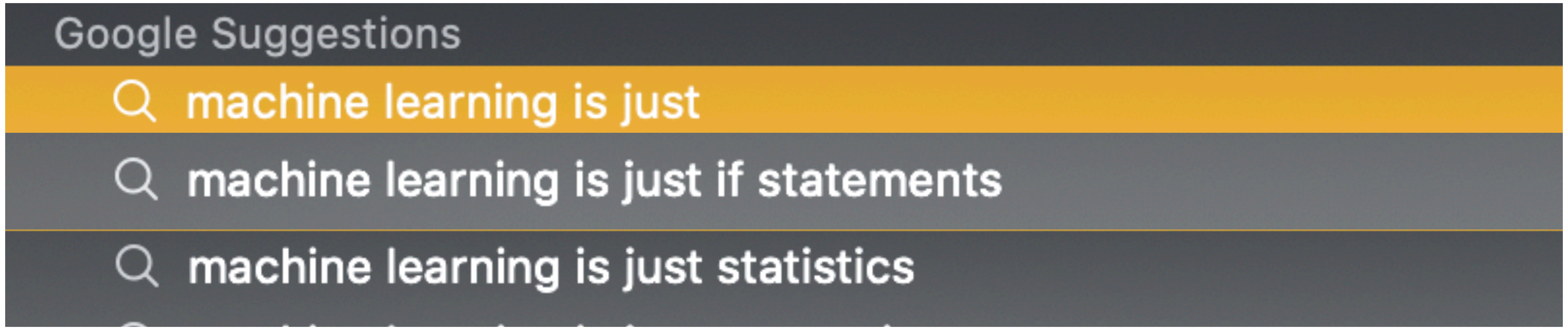


$$y(x, \mathbf{w}) = \mathbf{w} \cdot \mathbf{x} + b$$

$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

Plot of a training data set of $N = 10$ points, shown as blue circles, each comprising an observation of the input variable x along with the corresponding target variable t. The green curve shows the function $\sin(2\pi x)$ used to generate the data. Our goal is to predict the value of t for some new value of x, without knowledge of the green curve.





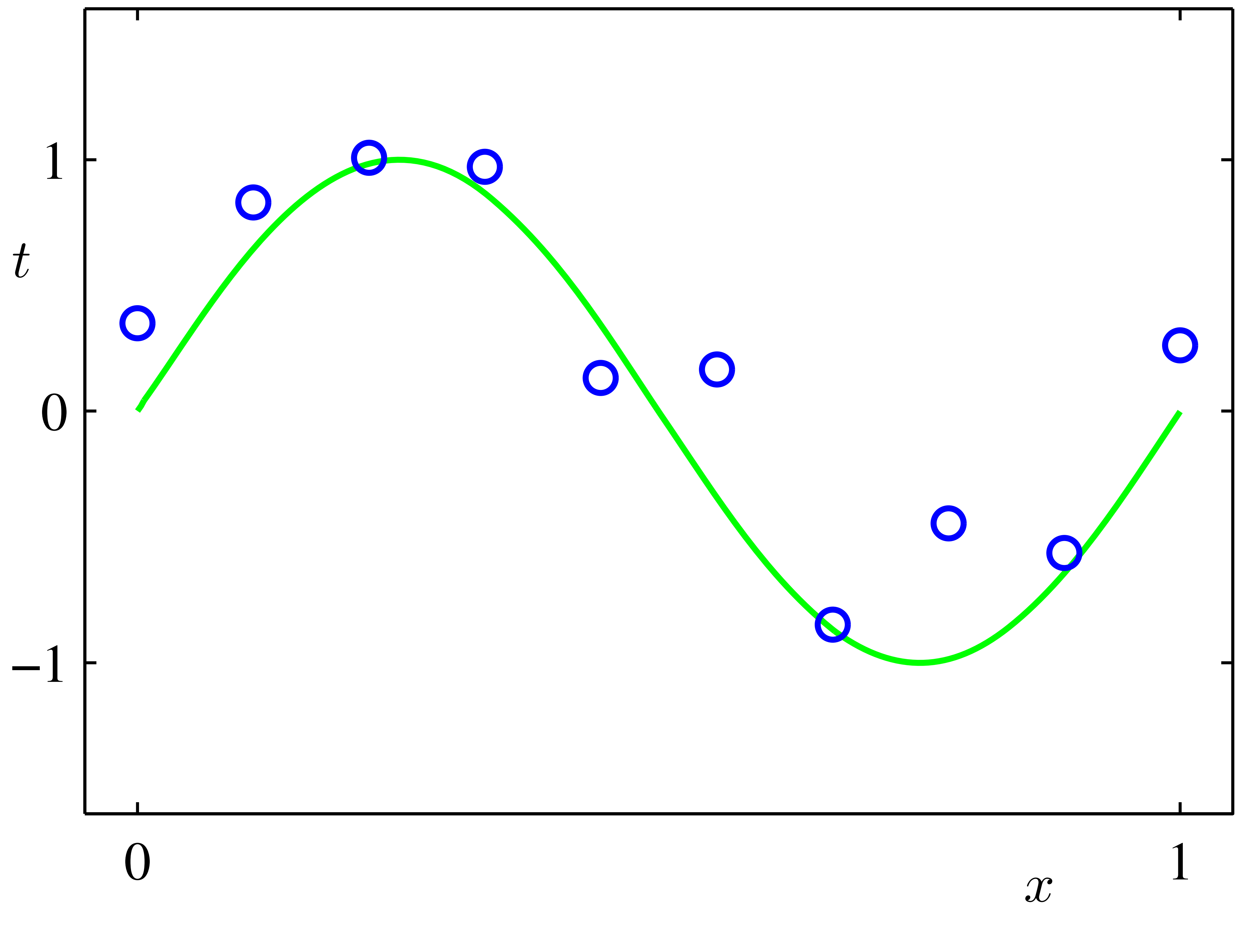
$$\text{if } (\%\texttt{george} < 0.6)\ \&\ (\%\texttt{you} > 1.5) \qquad \begin{array}{l}\text{then } \texttt{spam} \\ \text{else } \texttt{email}.\end{array}$$

TABLE 1.1. *Average percentage of words or characters in an email message equal to the indicated word or character. We have chosen the words and characters showing the largest difference between* `spam` *and* `email`*.*

	george	you	your	hp	free	hpl	!	our	re	edu	remove
spam	0.00	2.26	1.38	0.02	0.52	0.01	0.51	0.51	0.13	0.01	0.28
email	1.27	1.27	0.44	0.90	0.07	0.43	0.11	0.18	0.42	0.29	0.01

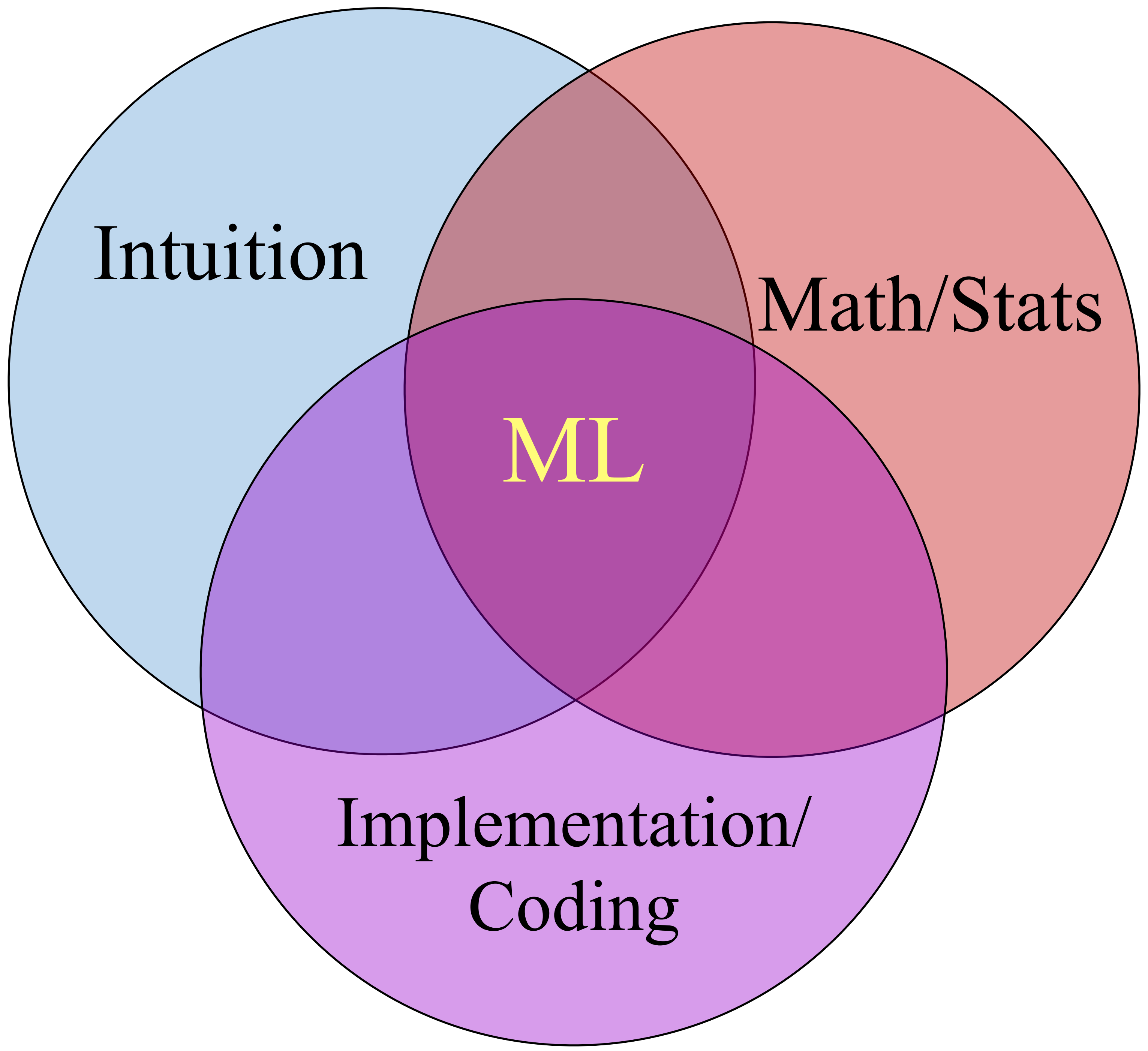
"If software ate the world, models will run it"
— Steven A. Cohen and Matthew W. Granade, The Wallstreet Journal, 2018

$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$



Challenges of ML

- Data
  - Insufficient
  - non-representative
  - poor quality
- Testing how good your model is
  - ML is too powerful - overfit!
  - Generalization error: the errors on new data not used in training
- Selecting a good enough/better/best ML
  - “Hyperparameter tuning”
  - “Model selection”
- How do you even measure performance? What’s the metric?

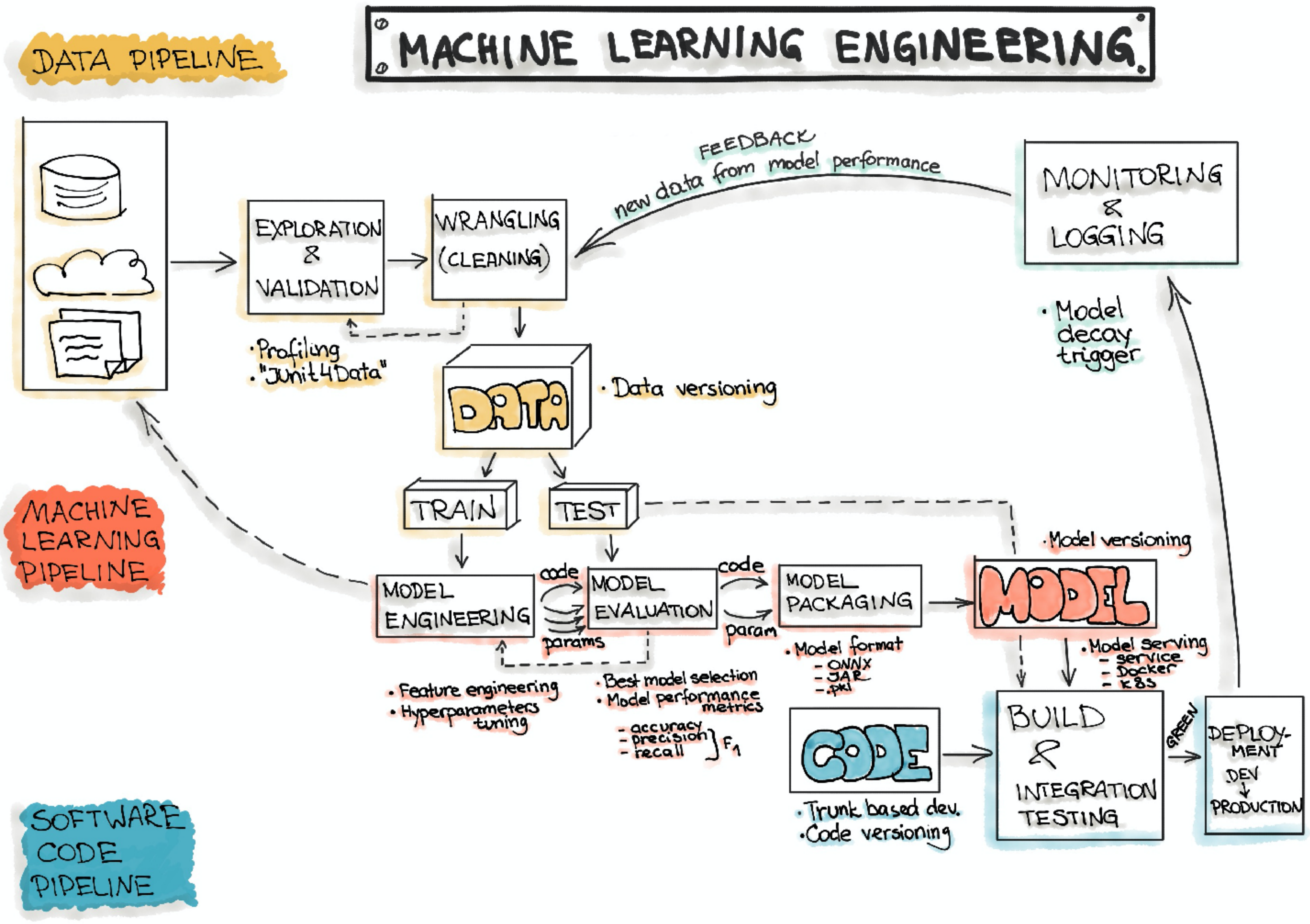


To do ML right you need all of these:
Intuition & Domain knowledge + Math/Stats
+ Implementation/Coding + GOOD DATA

The process and art of machine learning

(At a small scale)

- Acquire data and knowledge about the subject
- Curate & clean the data
- Explore the data
- Make useful transformations of the data
- Split the data in different ways according to what you need to do
    - e.g., Train/test set split, Cross-validation, Nested cross-validation
    - More complexity of model-building, Less data —> more elaborate data splitting
- “DO ML”
- Evaluate and interpret your results


MACHINE LEARNING ENGINEERING
DATA PIPELINE
MACHINE LEARNING PIPELINE
SOFTWARE CODE PIPELINE
EXPLORATION & VALIDATION
WRANGLING (CLEANING)
· Profiling
· "Junit4Data"
FEEDBACK
new data from model performance
MONITORING & LOGGING
· Model decay trigger
DATA
· Data versioning
TRAIN
TEST
MODEL ENGINEERING
code
params
MODEL EVALUATION
code
param
MODEL PACKAGING
MODEL
· Model versioning
· Feature engineering
· Hyperparameters tuning
· Best model selection
· Model performance metrics
- accuracy
- precision
- recall
F_1
· Model format
- ONNX
- JAR
- .pkl
· Model serving
- service
- Docker
- k8s
CODE
· Trunk based dev.
· Code versioning
BUILD & INTEGRATION TESTING
GREEN
DEPLOYMENT
DEV
PRODUCTION

ml-ops.org

Stopped here for time

Predict / classify or model?

Usually ML

Usually Stats

Basic notation

INPUT DATA

We use x (lower case) to denote a feature value (scalar).

The ith input data sample is represented as a vector using bold $\mathbf{x}$:

$\mathbf{x}_i = (x_{i1}, ..., x_{im}) \in \mathbb{R}^m$: A row vector of m elements.

$$\mathbf{x}_i = (22, 1, 0, 160, 180)$$

The entire dataset is represented by a set (the sequence in which each data input $\mathbf{x}_i$ usually doesn't matter.

$S = \{\mathbf{x}_i, i = 1..n\}$: A set S with n samples. i goes from 1 to n.

Or we can write it as a matrix, when we need to do some linear algebra :)

Basic notation

PREDICTION

We use y (lower case) to denote a binary classification.

$y = -1$ (or sometimes we use $y = 0$) is referred to as the negative class.
$y = +1$ is referred to as the positive class.

Given a data sample $\mathbf{x}_i = (x_{i1}, ..., x_{im})$,

we want to predict $y_i = -1 \; or \; +1$?

OR… y is just a real number we want to predict

Given a data sample $\mathbf{x}_i = (x_{i1}, ..., x_{im})$,

we want to predict $y_i \in \mathbb{R}$?

Basic notation

MODEL PARAMETERS

Model: $\mathbf{w} = (w_1, ..., w_m) \in \mathbb{R}^m$ (in the same dimension of input $\mathbf{x}$)

bias: $b \in \mathbb{R}$ (scalar)

Data sample $\mathbf{x} = (x_1, ..., x_m) \in \mathbb{R}^m$,

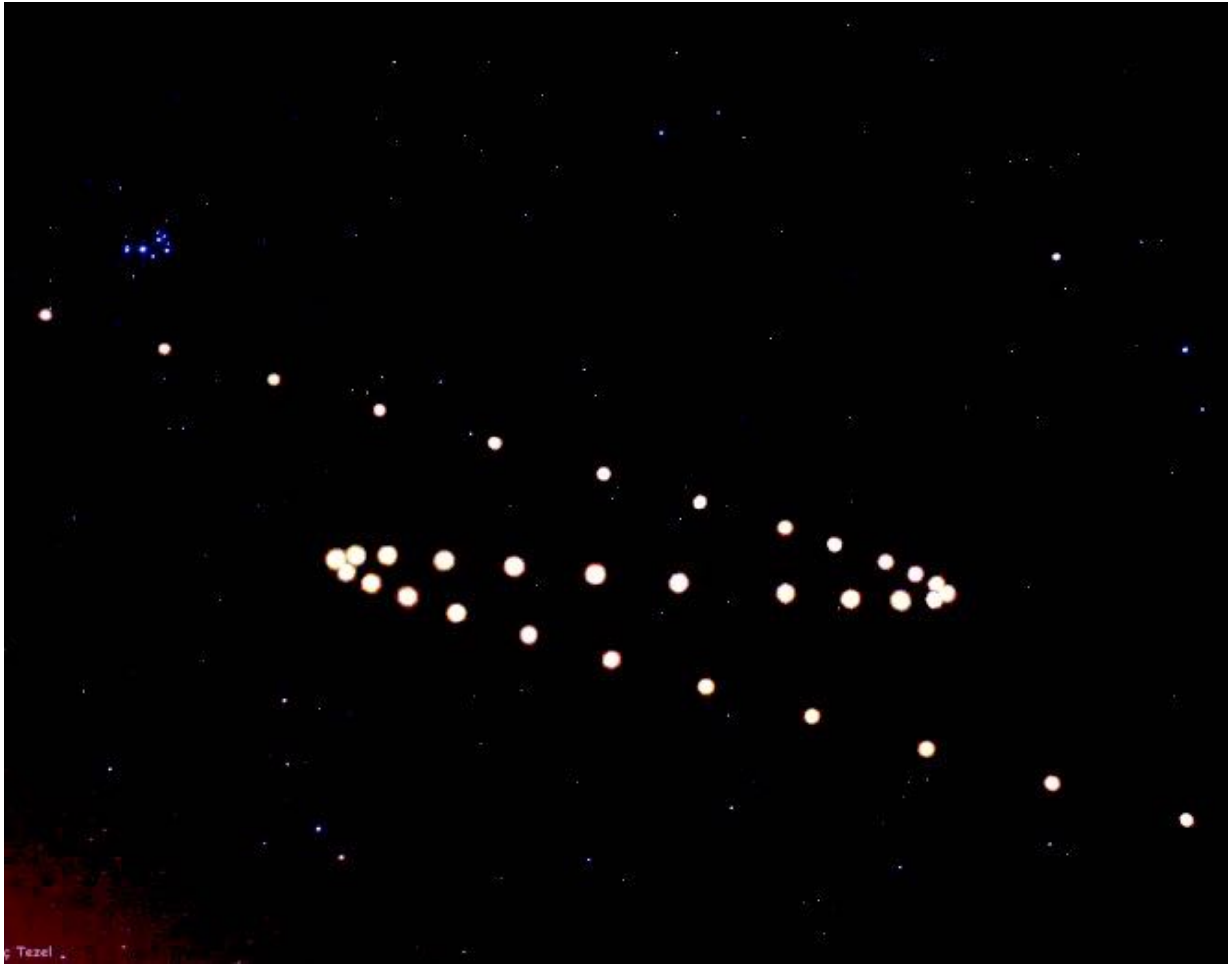
$$\mathbf{w} \cdot \mathbf{x} + b \qquad (w_1, w_2, .., w_m) \begin{pmatrix} x_1 \\ x_2 \\ . \\ . \\ x_m \end{pmatrix} + b$$

"$\cdot$" refers to as the dot product between two vectors

Alternative notation 1: $< \mathbf{w}, \mathbf{x} > +b$

Alternative notation 2: $\mathbf{w}\mathbf{x}^T + b$ ($\mathbf{w}$ and $\mathbf{x}$ are row vectors).

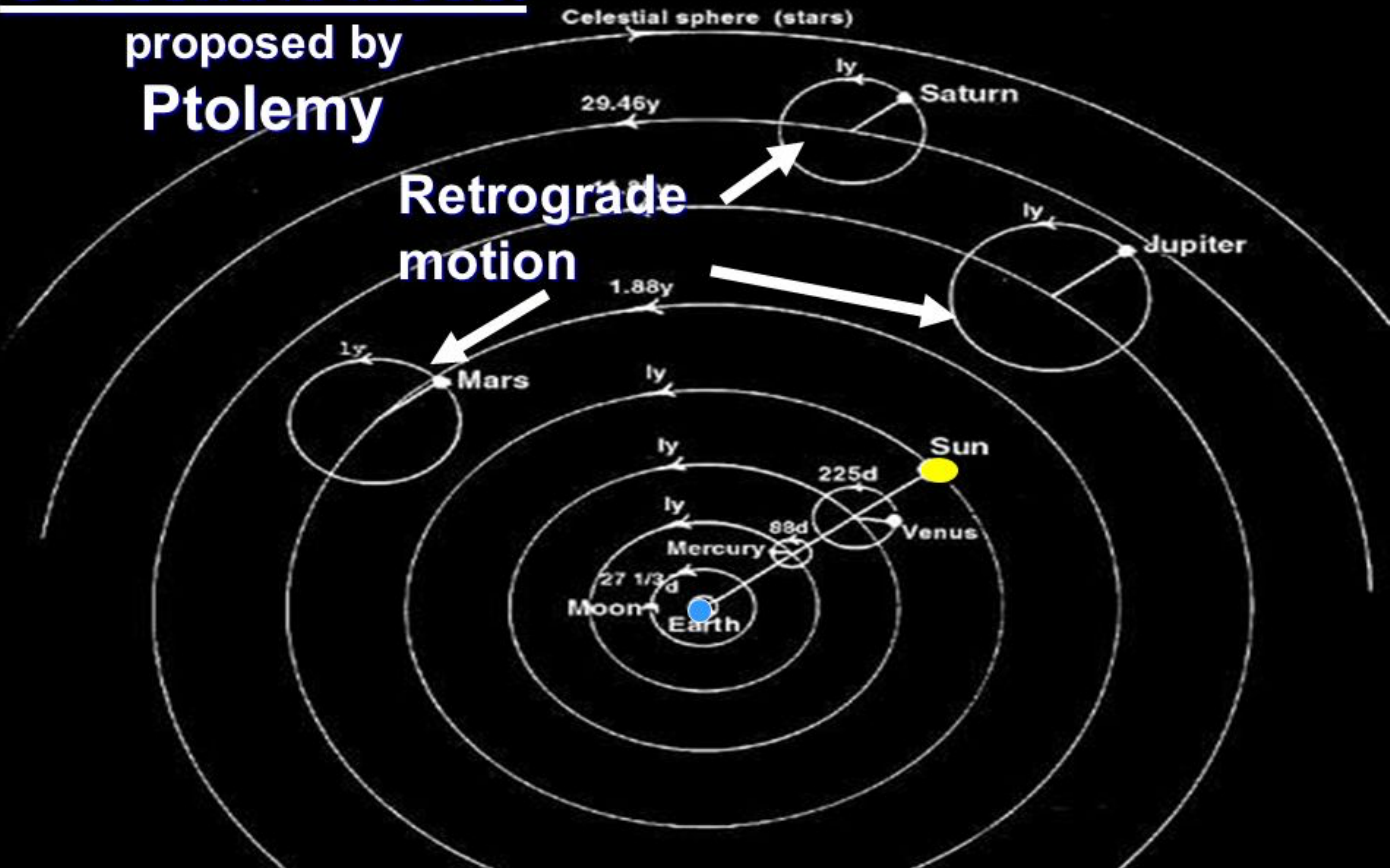
$\mathbf{w}^T\mathbf{x} + b$ ($\mathbf{w}$ and $\mathbf{x}$ are column vectors).


Tezel

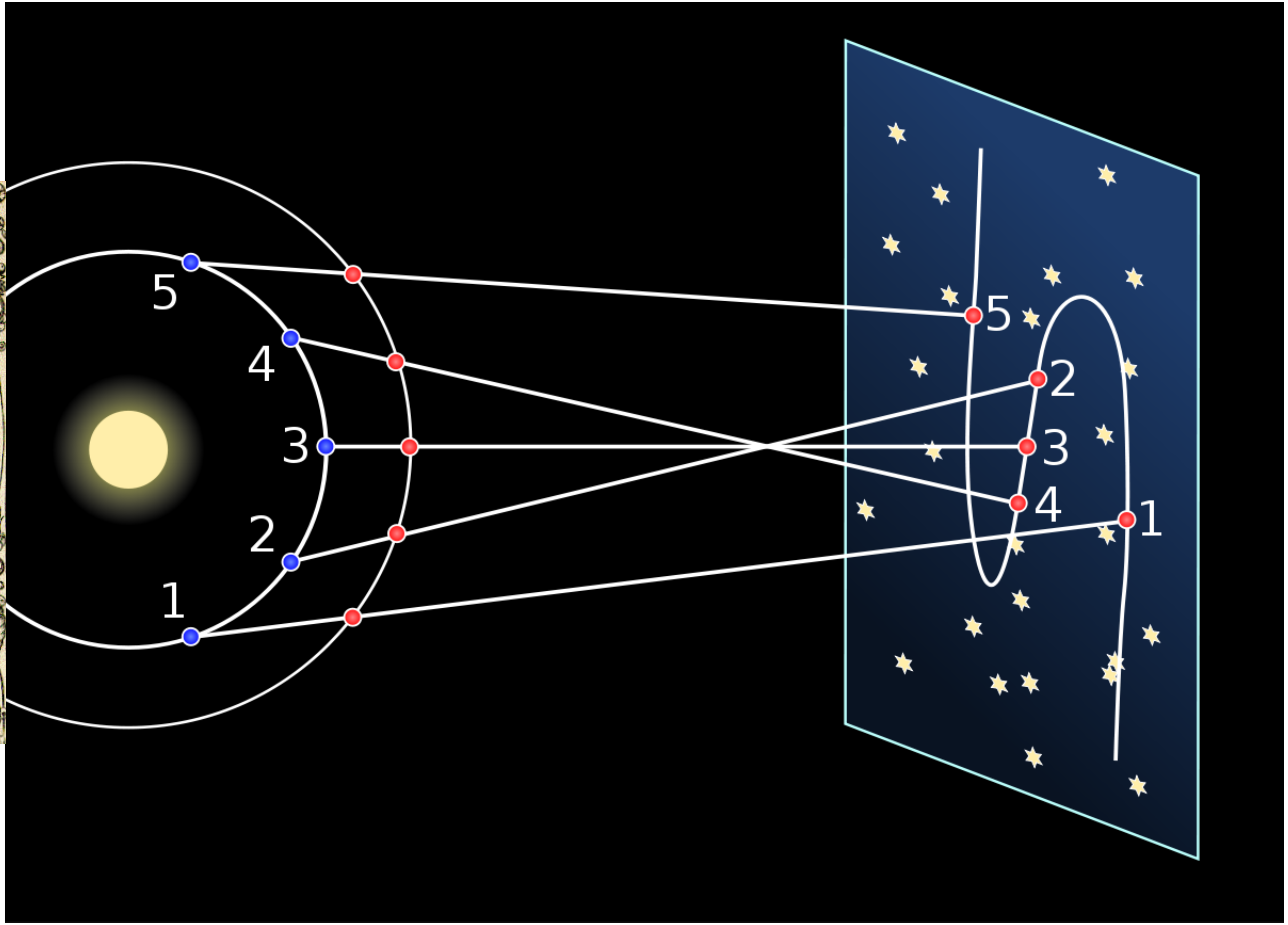
Geocentric model

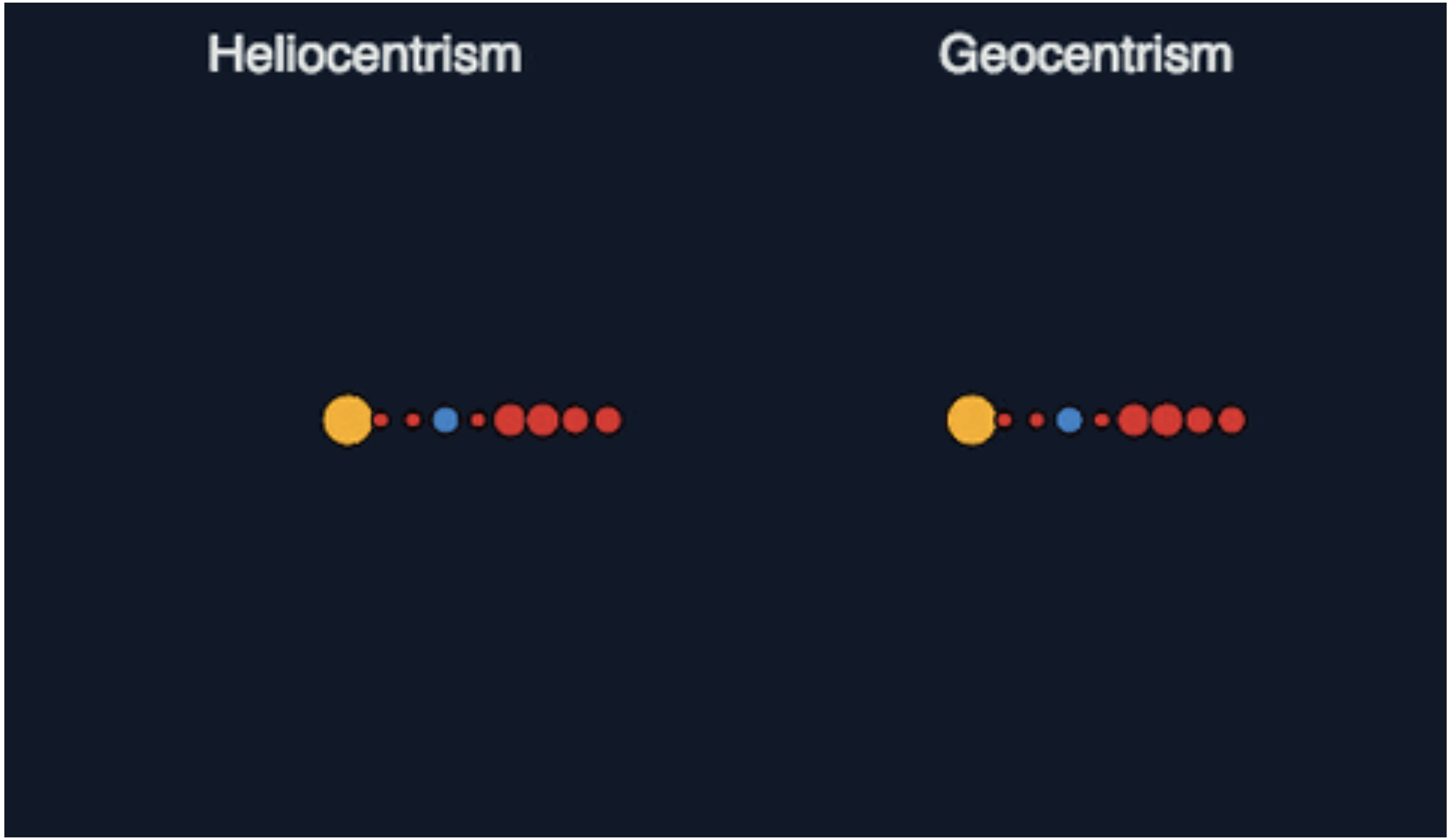
proposed by

Ptolemy




NICOLAUS COPERNICUS Tornaeus Borussus, Mathemat. Nat. A°. 1473. Ob. 1543.
Non docet instabiles Copernicus aetheris orbes,
Sed terrae instabiles arguit ille vices.

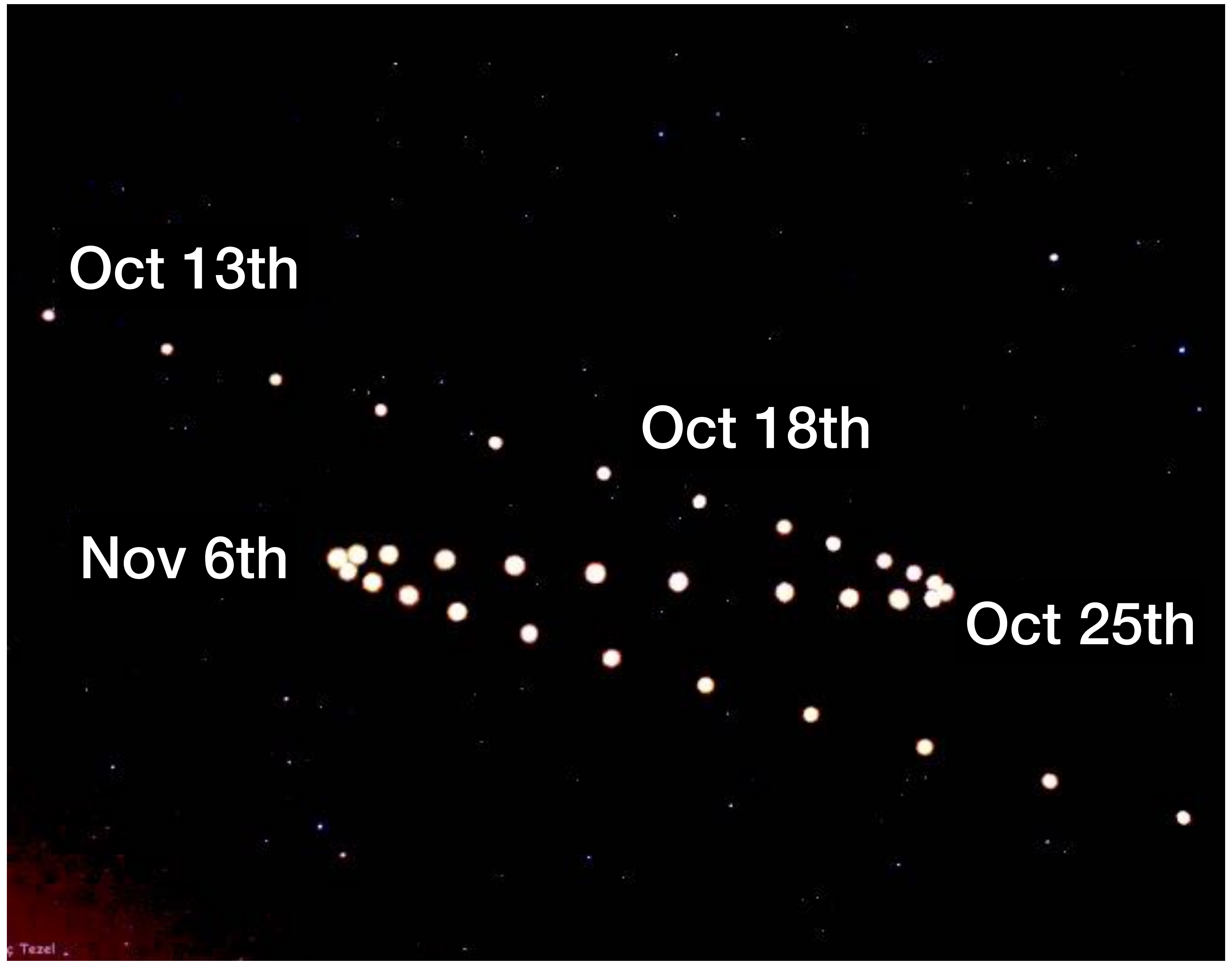

1
2
3
4
5


Heliocentrism
Geocentrism

What is the goal?

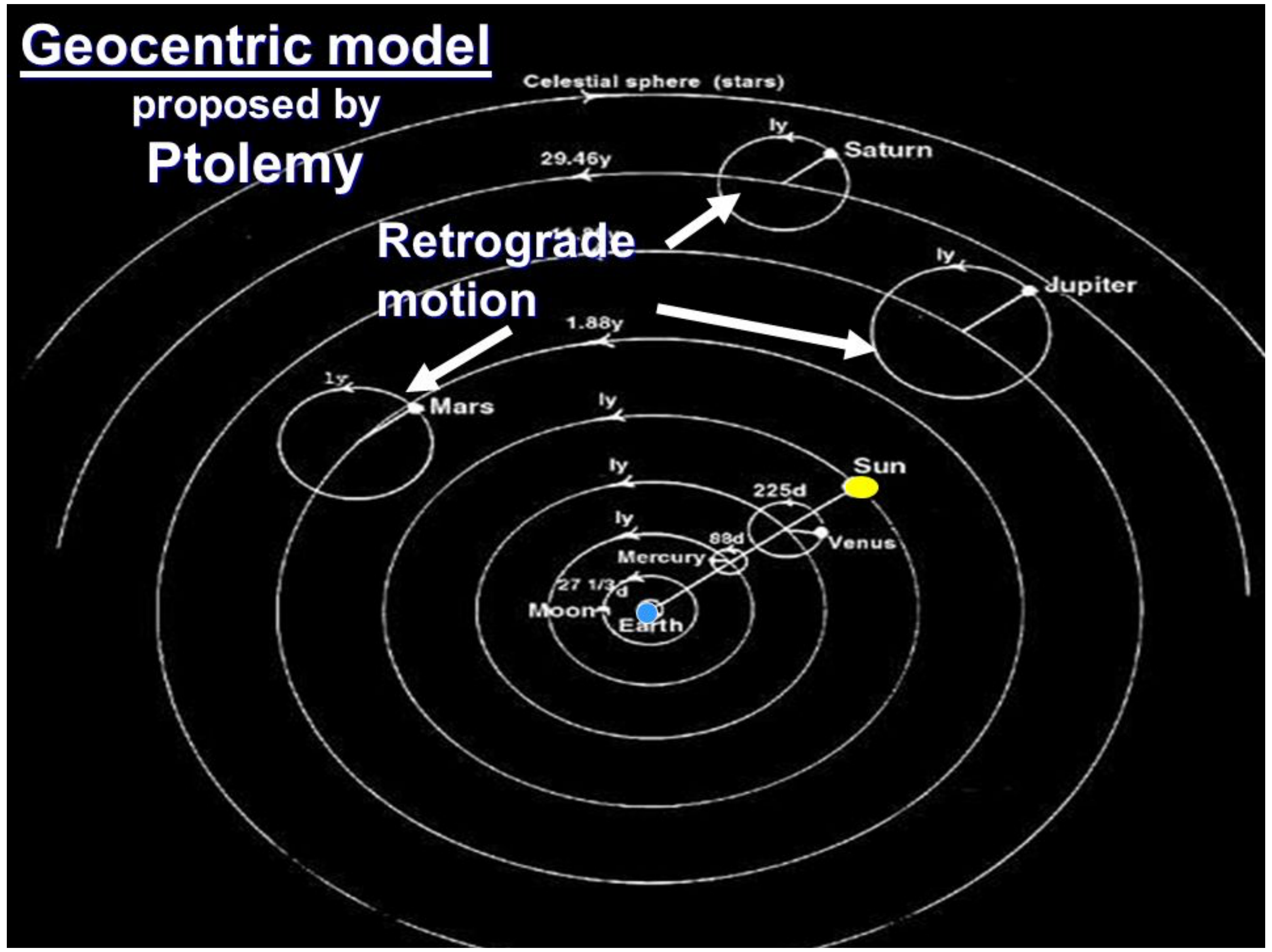
$$\mathbf{y} = f(\mathbf{w}; \mathbf{x})$$

Prediction



Only care that prediction y has low error

Modeling



We care that model w is an accurate representation of the real thing