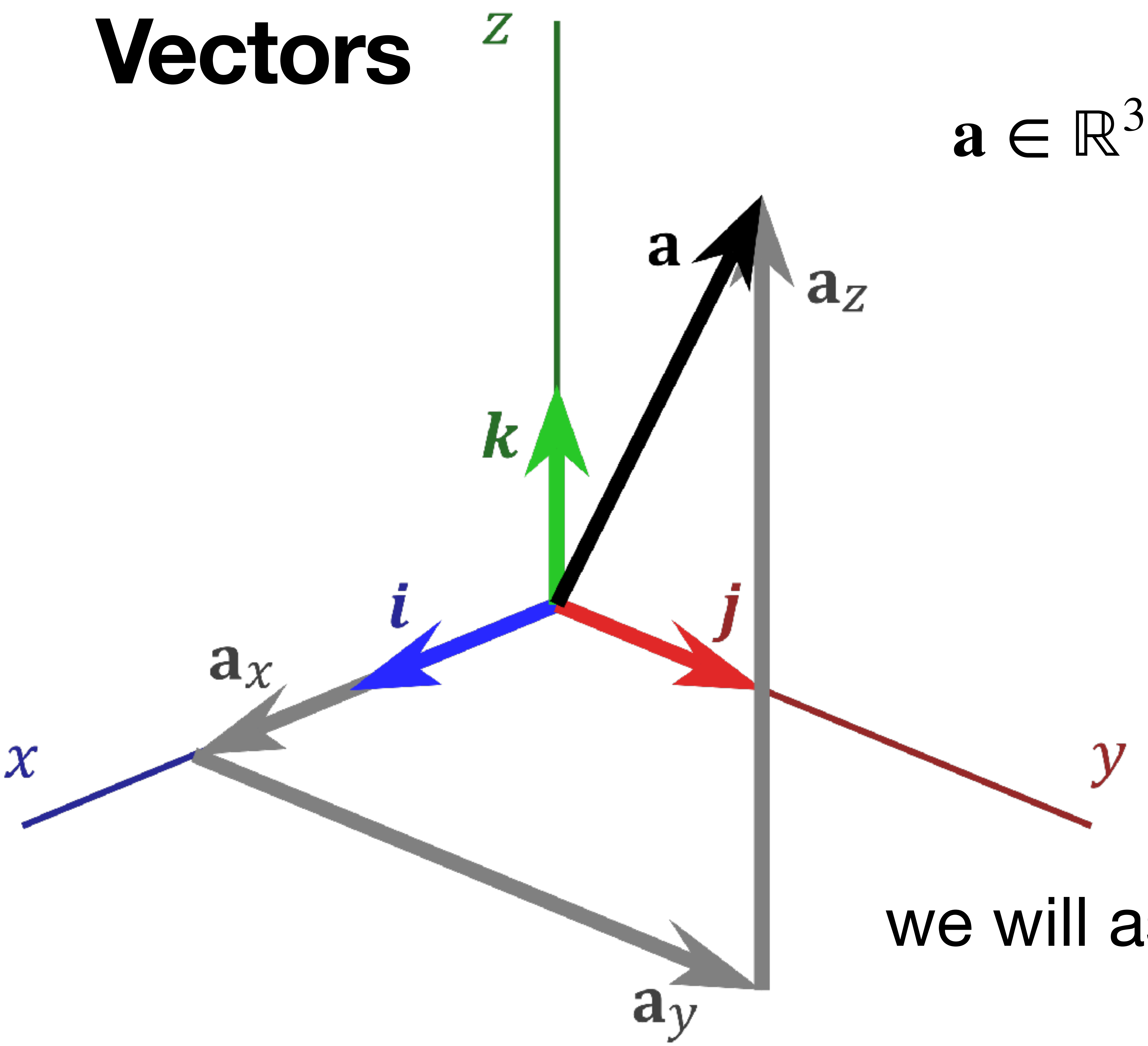
Lecture 2 pre-video

Everything in ML is a vector or matrix

Vector operations we need

- Vector addition
- Vector multiplication
  - Vector with scalar
  - Between two vectors to produce a scalar (dot product)
  - ~~Between two vectors to produce a vector (cross product)~~

Vectors



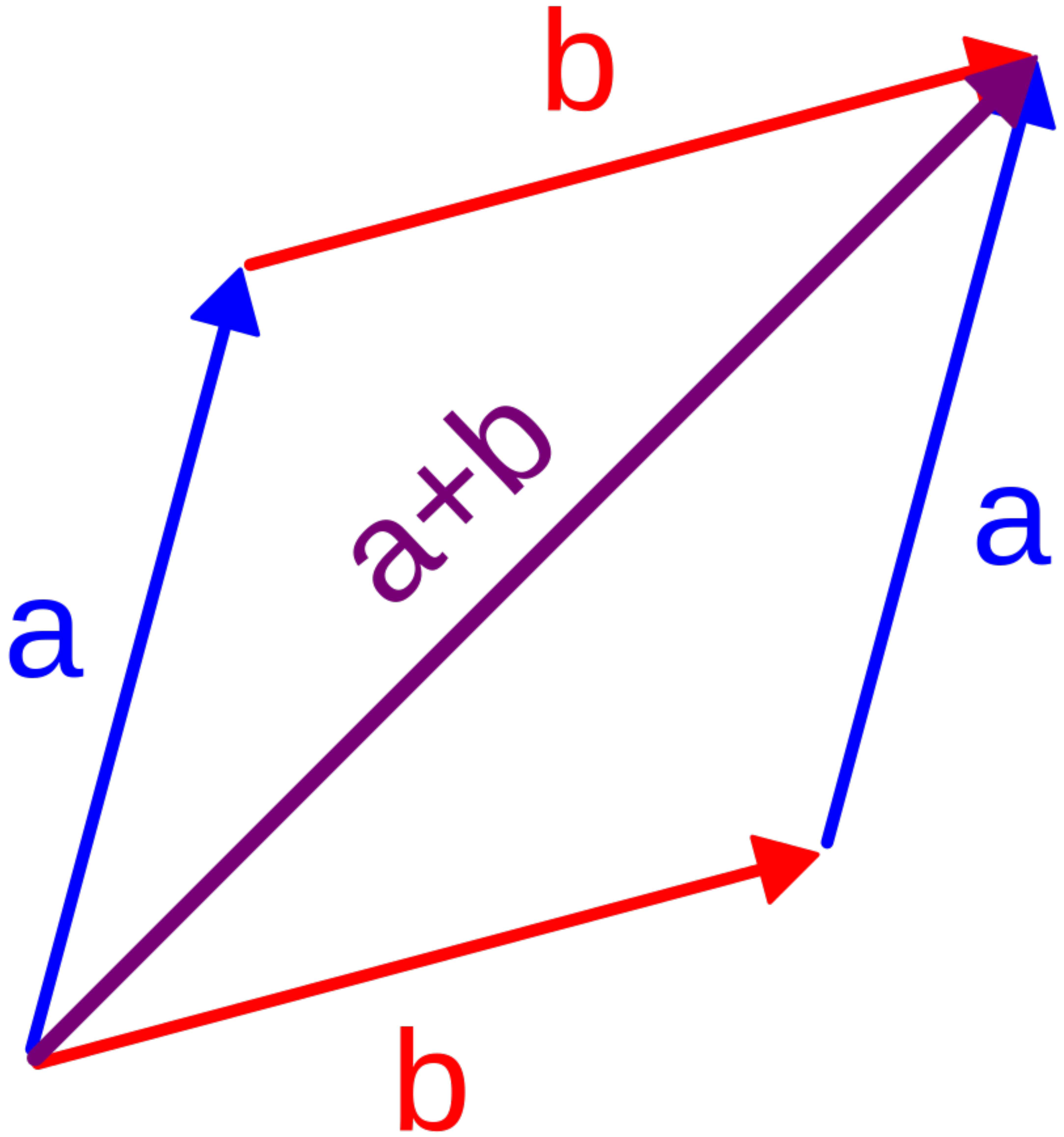
$$\mathbf{a} \in \mathbb{R}^3 \qquad \mathbf{a} = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}$$

$$\|\mathbf{a}\|_2 = \sqrt{a_x^2 + a_y^2 + a_z^2}$$

$$\|\mathbf{a}\|_n = \left(a_x^n + a_y^n + a_z^n\right)^{1/n}$$

we will assume that $\|\mathbf{a}\|$ means $\|\mathbf{a}\|_2$

Vector addition

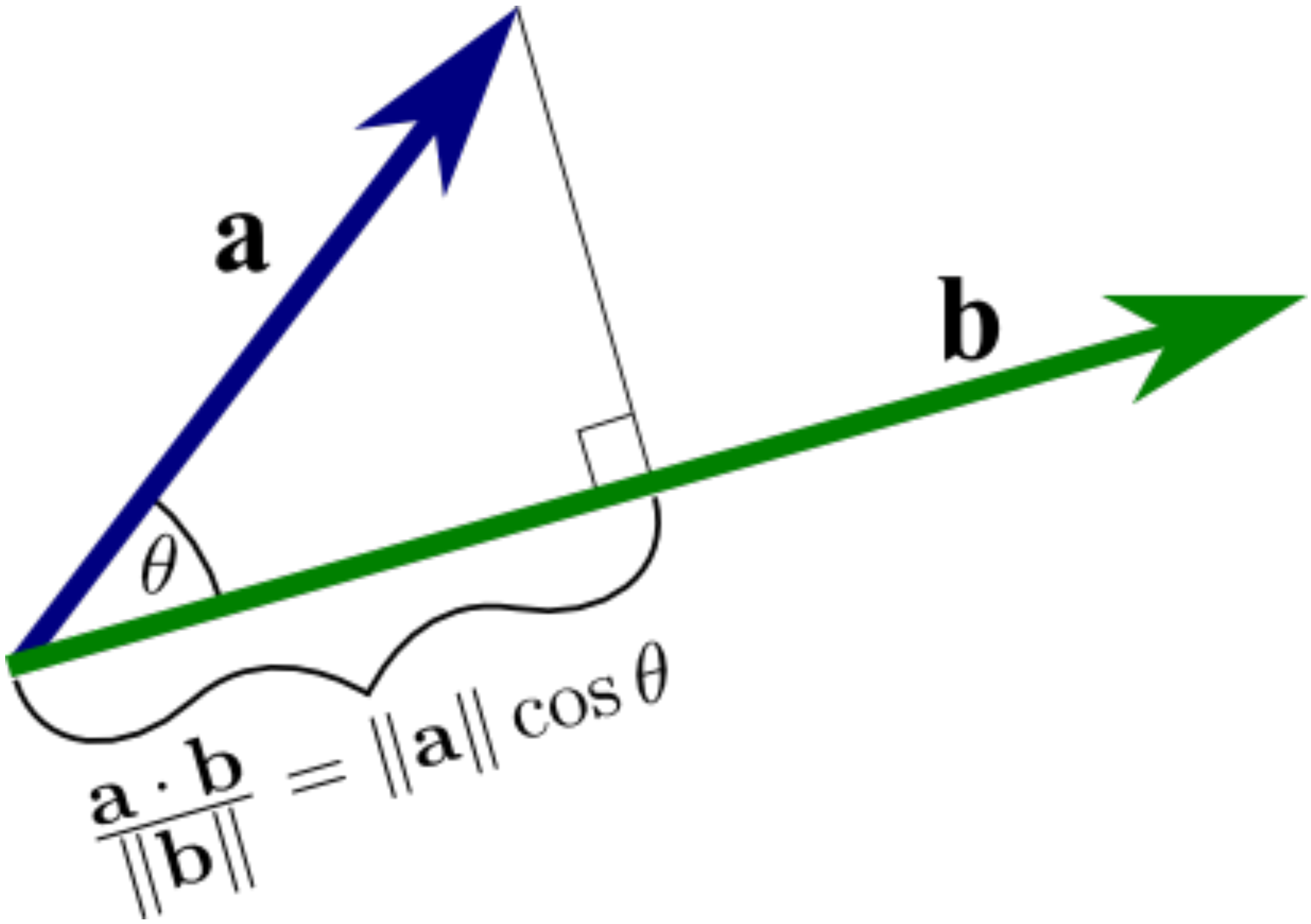


$$\mathbf{a}, \mathbf{b} \in \mathbb{R}^2$$

$$\mathbf{a} = \begin{bmatrix} a_x \\ a_y \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} b_x \\ b_y \end{bmatrix}$$

$$\mathbf{a} + \mathbf{b} = \begin{bmatrix} a_x + b_x \\ a_y + b_y \end{bmatrix}$$

Dot product - a scalar projection



$$\mathbf{a} \cdot \mathbf{b} = \|\mathbf{a}\| \, \|\mathbf{b}\| \, \cos\theta$$

$$\mathbf{a} \cdot \mathbf{b} \equiv \langle \mathbf{a}, \mathbf{b} \rangle$$

$$\mathbf{a} \cdot \mathbf{b} \equiv \mathbf{a}^{\mathrm{T}} \mathbf{b}$$

Matrix multiplication

Vector:

$$A = \begin{pmatrix} a_1 & a_2 & a_3 \end{pmatrix} \qquad B = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

$$AB = \begin{pmatrix} a_1 & a_2 & a_3 \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = a_1b_1 + a_2b_2 + a_3b_3$$

$$AB \neq BA$$

$$BA = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} \begin{pmatrix} a_1 & a_2 & a_3 \end{pmatrix} = \begin{pmatrix} b_1a_1 & b_1a_2 & b_1a_3 \\ b_2a_1 & b_2a_2 & b_2a_3 \\ b_3a_1 & b_3a_2 & b_3a_3 \end{pmatrix}$$

Matrix multiplication

Matrix:

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{pmatrix} \qquad B = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \end{pmatrix}$$

$$AB = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{pmatrix} \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \end{pmatrix}$$

$$= \begin{pmatrix} a_{11}b_{11} + a_{12}b_{21} + a_{13}b_{31} & a_{11}b_{12} + a_{12}b_{22} + a_{13}b_{32} \\ a_{21}b_{11} + a_{22}b_{21} + a_{23}b_{31} & a_{21}b_{12} + a_{22}b_{22} + a_{23}b_{32} \end{pmatrix}$$

Lecture #2

How similar are two data points?

Jason G. Fleischer, Ph.D.
Asst. Teaching Professor
Department of Cognitive Science, UC San Diego

jfleischer@ucsd.edu

@jasongfleischer

https://jgfleischer.com

Logistics

- Things to do this week:
  - Do syllabus quiz on Canvas
  - Do practice assignment on Datahub
  - Watch the vids / read optional things before lecture
  - No section!
  - Daily lecture survey (see syllabus!)
- Datahub is live for this class! You can also explore
  - Google Colab for additional free computation
  - Installing your own Anaconda

Python resources for you

- https://swcarpentry.github.io/python-novice-inflammation/ is a good intro to Python for people who will be using it to handle data. Uses numpy instead of pandas; covers matplotlib
- COGS108 will get you up to speed with all data wrangling (including numpy, pandas, matplotlib) you could possibly need
  - notebooks: https://github.com/COGS108/Tutorials
  - last quarter's lectures: https://github.com/COGS108/Resources
- A more in depth alternative to COGS108 is the free Python Data Science Handbook (includes Colab notebooks)
- Need to look up something you kinda know how to do, but don't remember exactly how?
  - https://chrisalbon.com
  - https://pandas.pydata.org/Pandas_Cheat_Sheet.pdf
  - https://github.com/rougier/matplotlib-cheatsheet/blob/master/matplotlib-cheatsheet.pdf

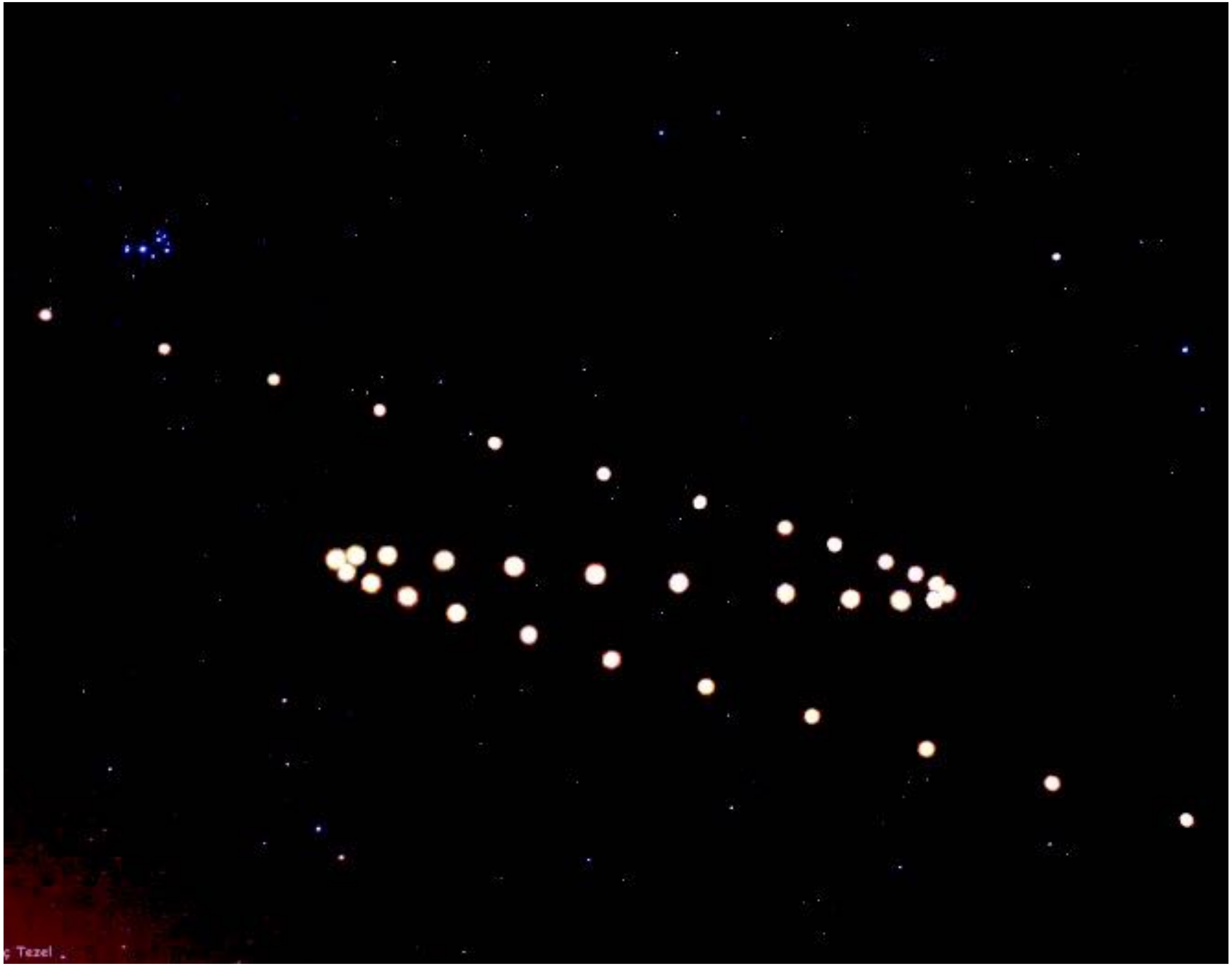
Running python RIGHT NOW

- Option #1 (easy) Get started with Google Colaboratory
    - Here's a Video tutorial series too
    - Good: Everything you need, for free, via your web browser and google drive
    - Limitations: If you are very ambitious in your project you might find the free instance limiting in memory or speed. Maybe, but unlikely.
- Option #2 (harder) Install Anaconda on your machine … there's a video tutorial about it at the installation page
    - Good: Everything you need, for free, on your machine in your control
    - Bad: Need to learn how to handle Anaconda, responsible for maintaining and upgrading the packages you use (will inevitably cause headaches, but you gotta learn sometime I guess?)
    - Limitations: How good is your hardware and sysadmin skill?

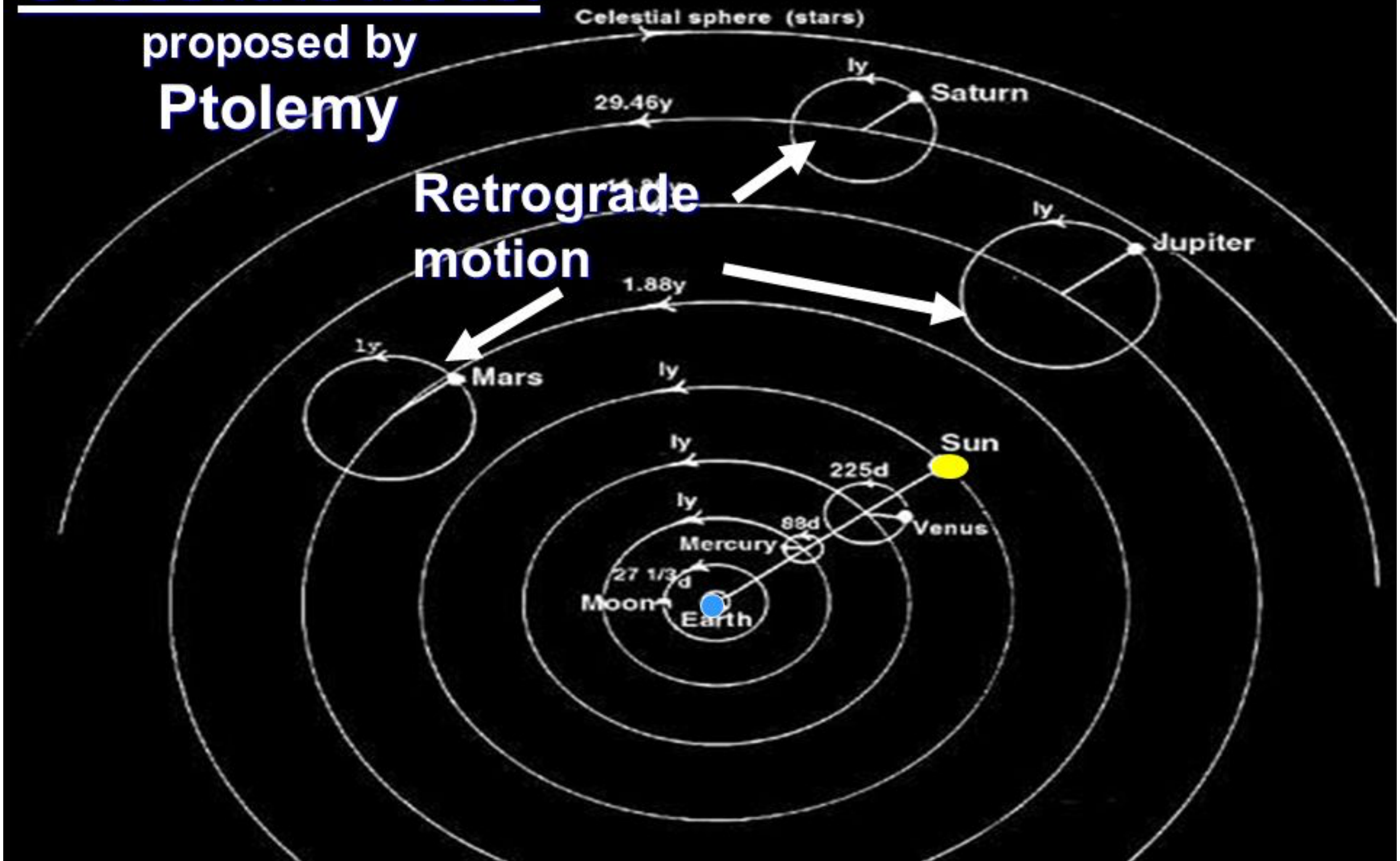
Predict / classify or model?

Usually ML

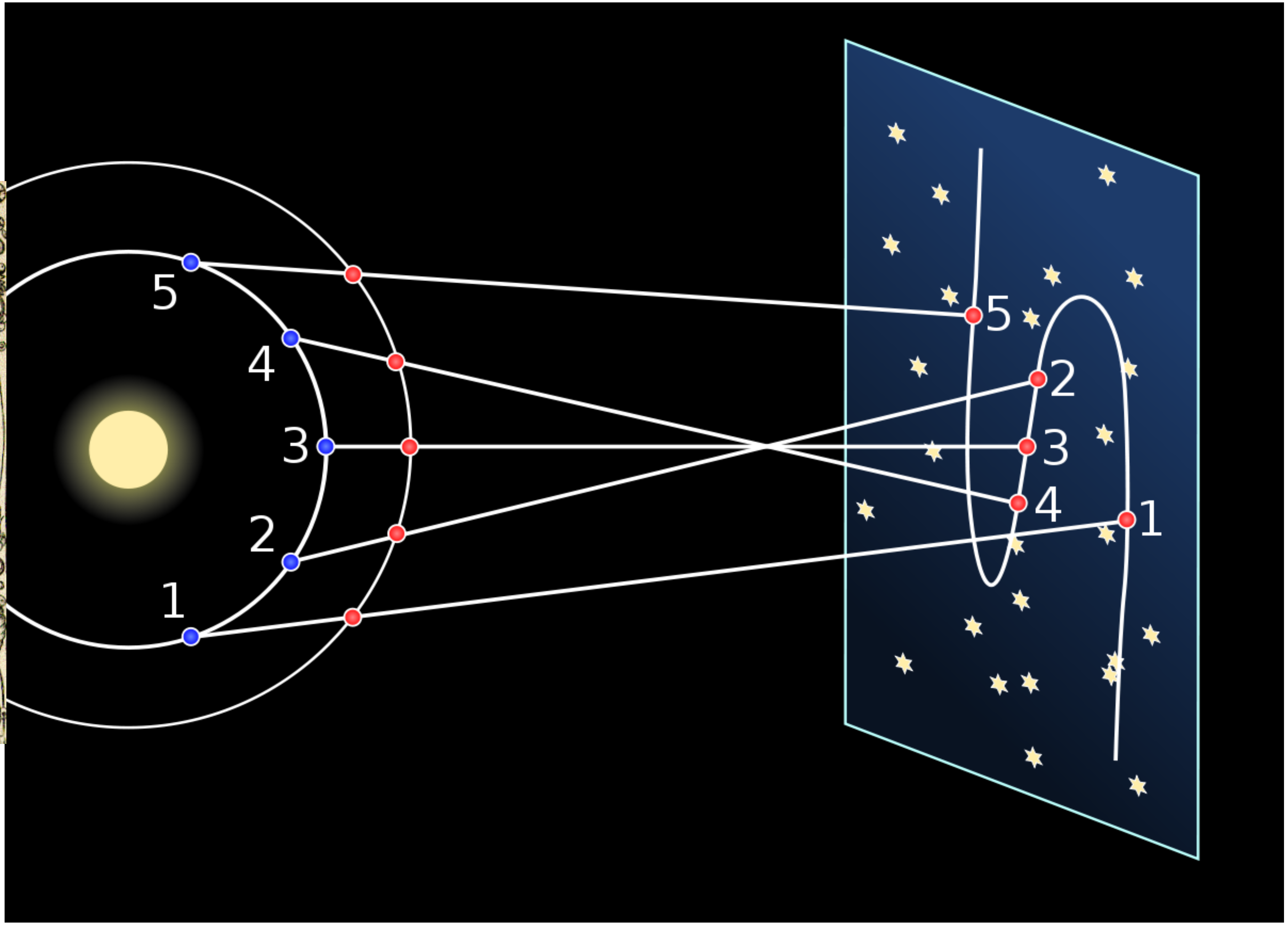
Usually Stats

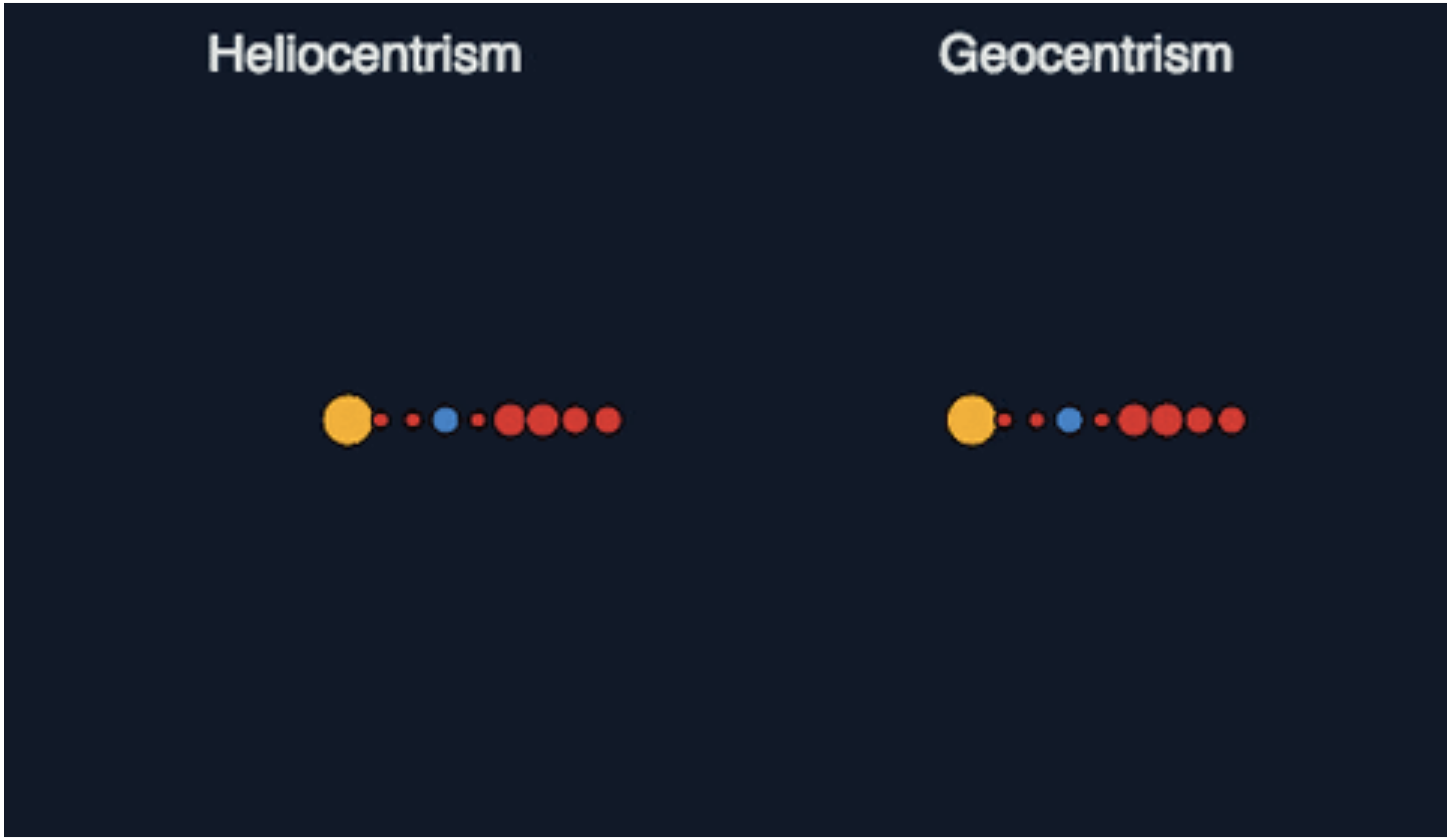

Tezel

Geocentric model
proposed by
Ptolemy




NICOLAUS COPERNICUS


5
4
3
2
1
5
2
3
4
1


Heliocentrism
Geocentrism

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

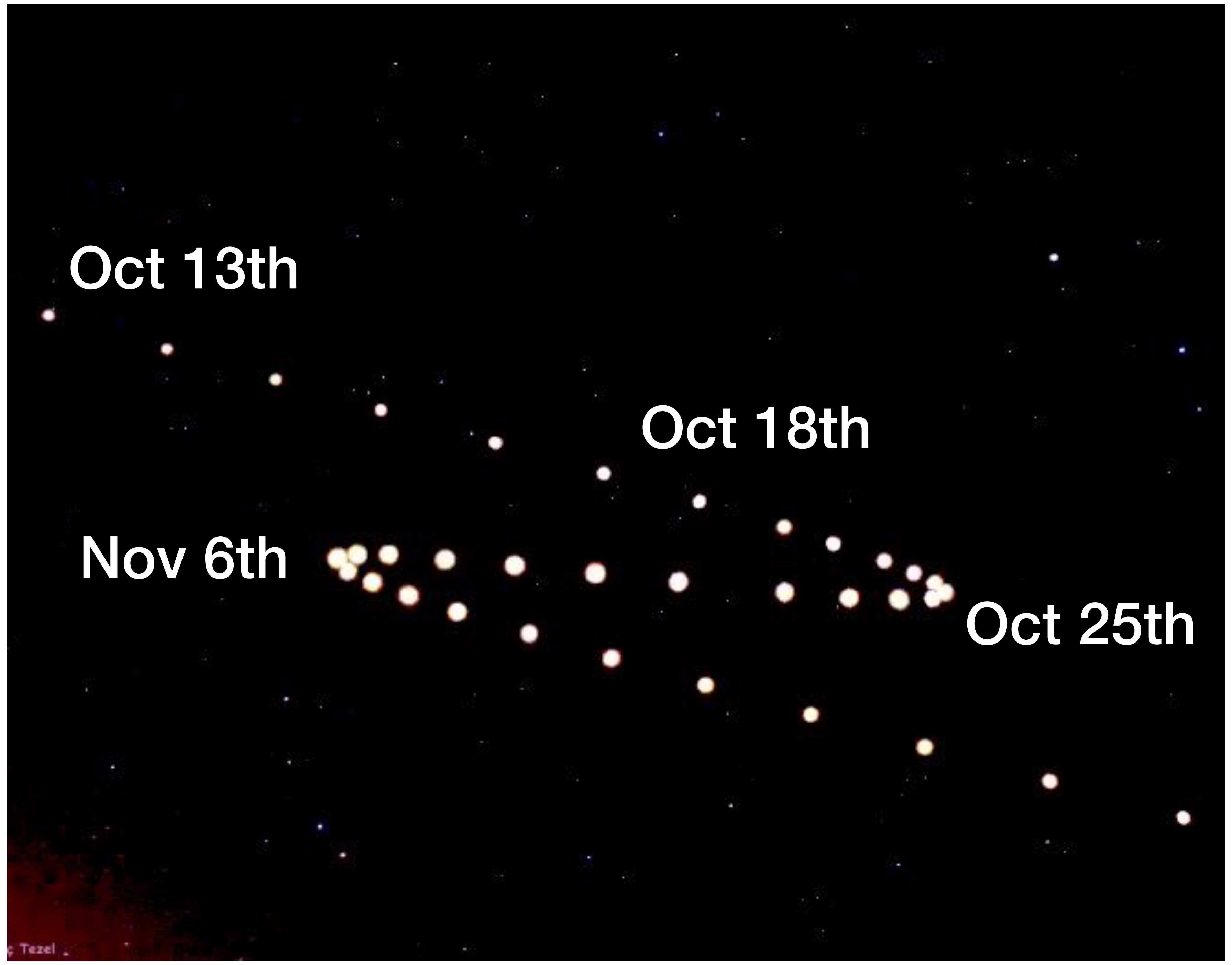
Alternative notation 1: $< \mathbf{w}, \mathbf{x} > +b$

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).

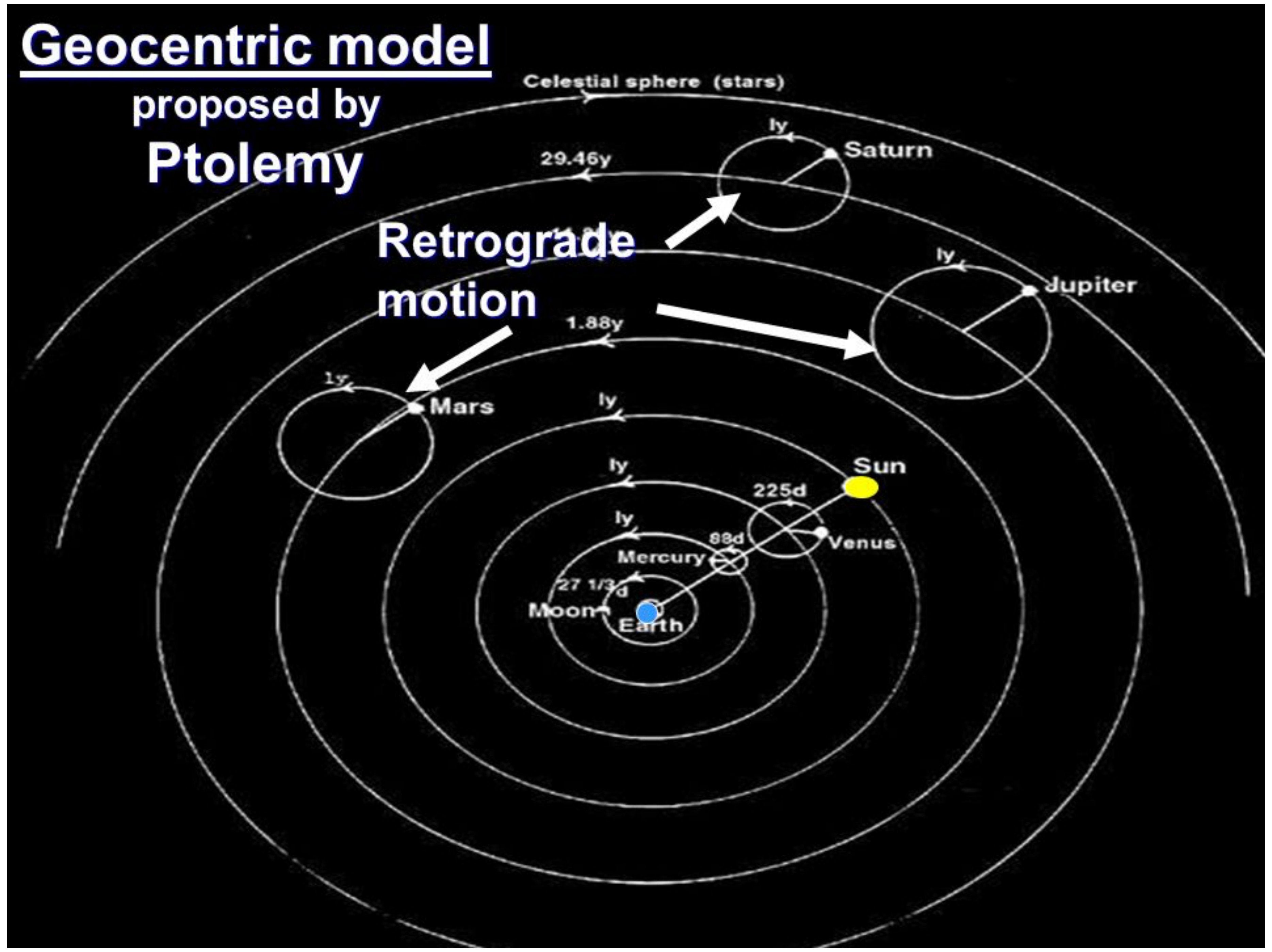
What is the goal?

Prediction

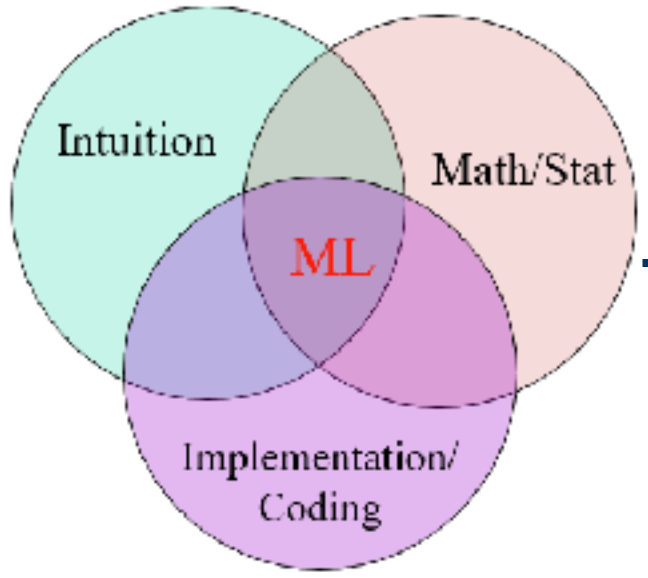


Only care that prediction
y has low error

Modeling



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



Recap: Supervised Learning

Intuition: A prediction task with a clear objective (e.g. a yes or no decision, which school to go to, a price to estimate, etc.) in which some history data for training can be acquired with the known prediction results already.

Math:

Training: $S_{training} = \{(\mathbf{x}_i, y_i), i = 1..n\}$

Testing: $S_{testing} = \{(\mathbf{x}_i), i = 1..u\}, what\ is\ y_i?$

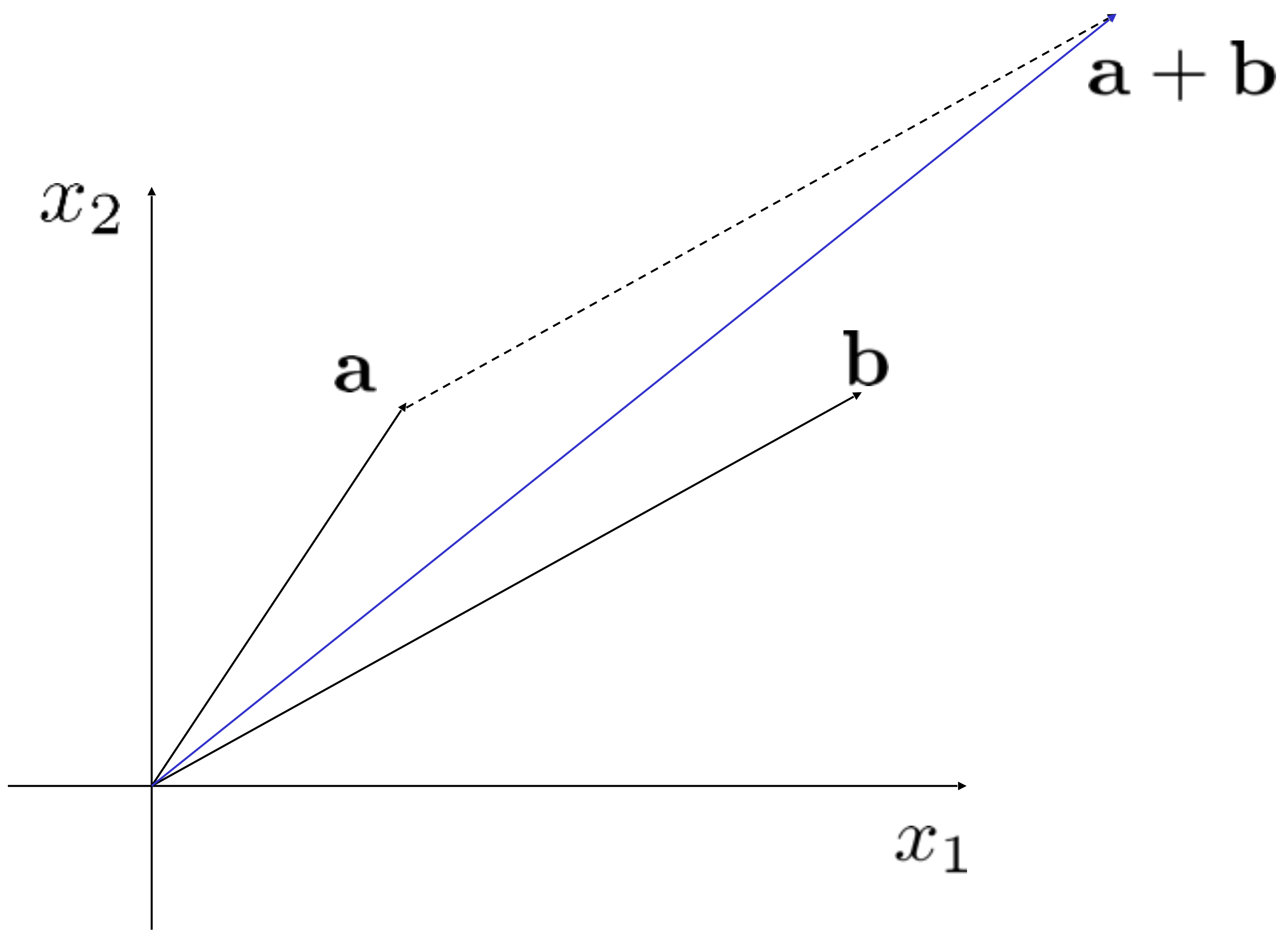
Linear algebra review on Canvas

See 3Blue1Brown if you want a much better refresher

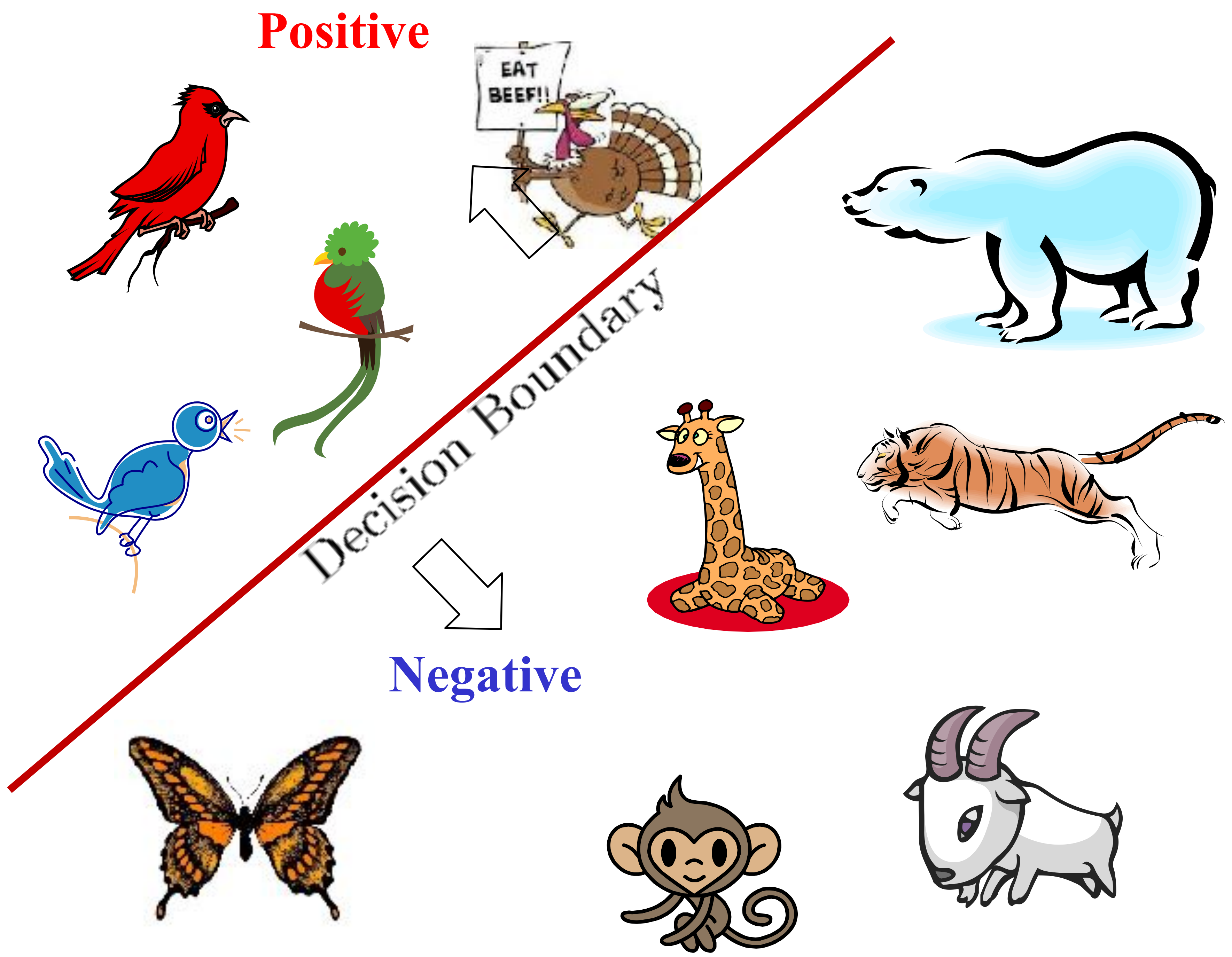
https://www.youtube.com/watch?v=fNk_zzaMoSs

Vector

Addition:



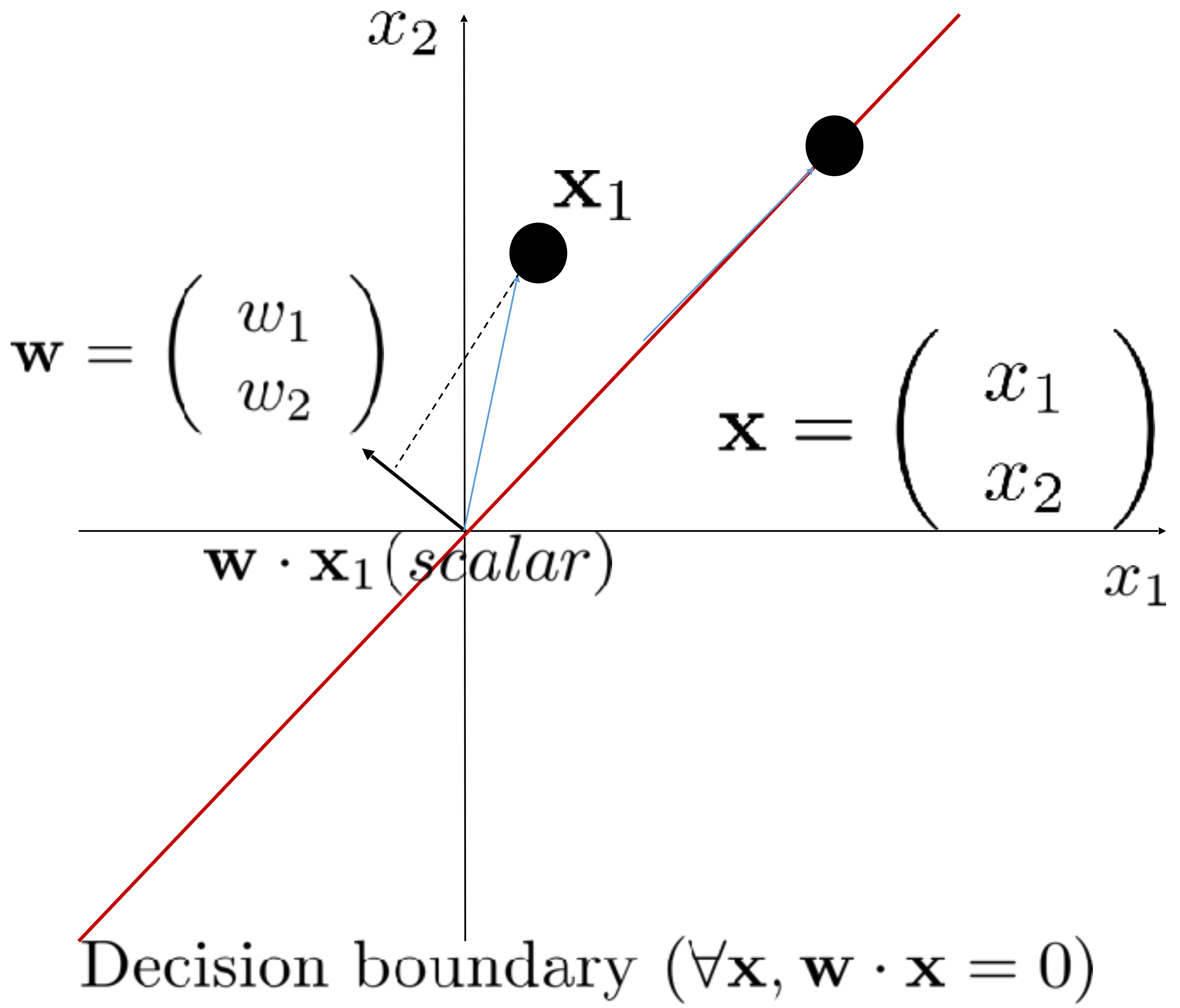
$$\mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \qquad \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$$



$$\mathbf{a}+\mathbf{b} = \begin{pmatrix} a_1 + b_1 \\ a_2 + b_2 \\ a_3 + b_3 \end{pmatrix}$$

It's still a vector in the same space as a and b.

Line and vector



Any point $\mathbf{x}$ on the line satisfies:

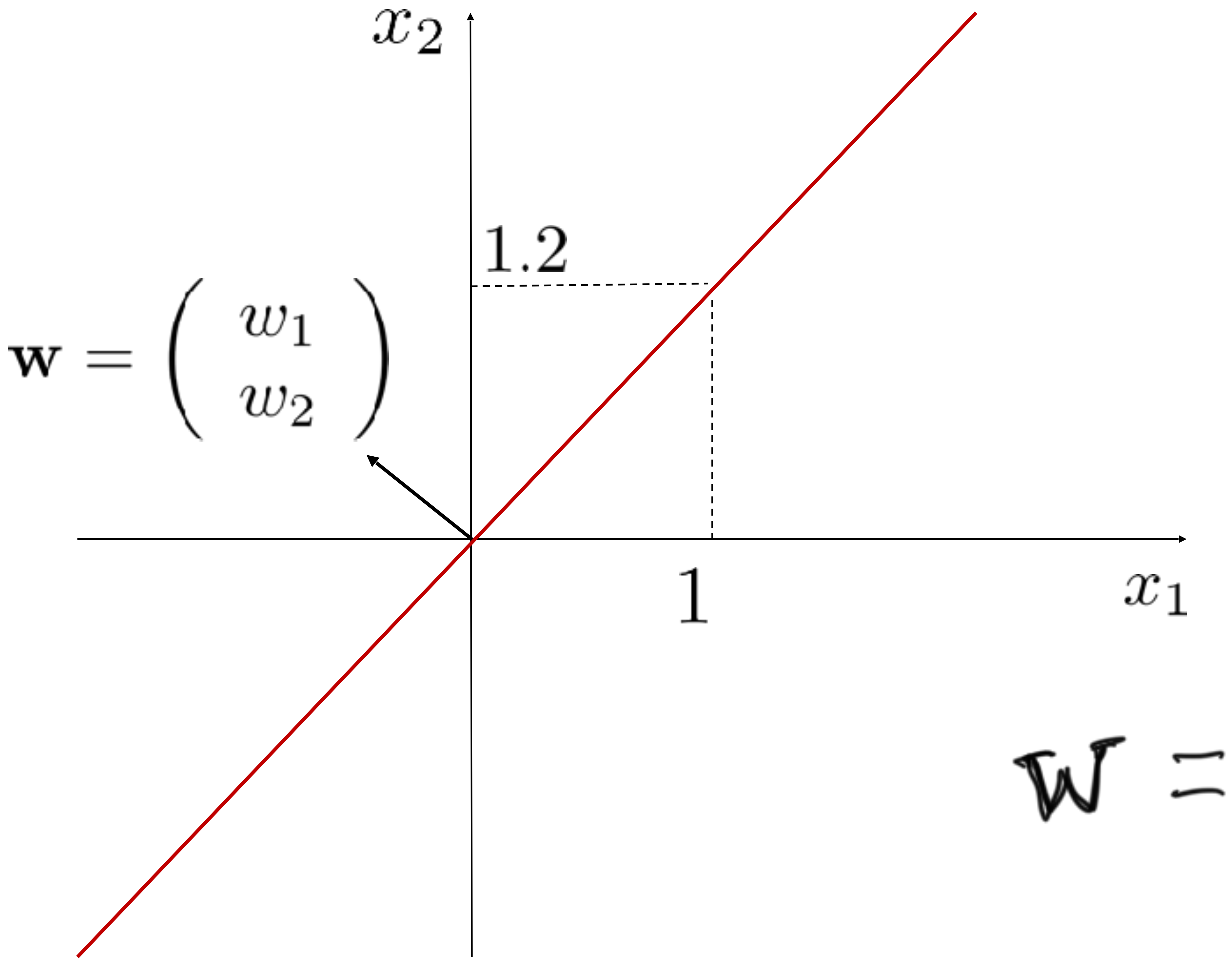
$$\mathbf{w}^T\mathbf{x} \equiv < \mathbf{w}, \mathbf{x} > \equiv \mathbf{w} \cdot \mathbf{x} = 0$$

$\mathbf{w}$ is the normal direction of the line

Often: $||\mathbf{w}||_2 = 1$: a unit vector

Line and vector

an example



$\mathbf{w} = \begin{pmatrix} \quad \end{pmatrix}$

$\mathbf{w}$ is the normal direction of the line

Often: $||\mathbf{w}||_2 = 1$: a unit vector

$||\mathbf{w}|| = 1 \Rightarrow \mathbf{w} = \begin{pmatrix} \quad \end{pmatrix}$

Significance of the dot product between two vectors

"Dot product" outputs a scalar value and it is arguably the most important mathematical operation in machine learning.

$$< \mathbf{a}, \mathbf{b} > \equiv \mathbf{a} \cdot \mathbf{b} \equiv \mathbf{a}^T \mathbf{b}$$
$$\equiv < \mathbf{b}, \mathbf{a} > \equiv \mathbf{b} \cdot \mathbf{a} \equiv \mathbf{b}^T \mathbf{a}$$

$$\mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \qquad \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

Why?

Computes the magnitude of the projection from one vector to the other, which measures the similarity between two vectors.

The dot product of two vectors is:

- largest when they are parallel
- 0 when they are orthogonal

The max value is $||a||\,||b||$… if vectors are unit length this is 1

Significance of the dot product between two vectors

	fly?	laying eggs?	weight (lb)
sparrow	yes	yes	0.087
chipmunk	no	no	0.19
bat	yes	no	0.09

Feature representation (one-hot encoded).

$$\text{sparrow} = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0.087 \end{pmatrix} \qquad \text{chipmunk} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0.19 \end{pmatrix} \qquad \text{bat} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0.09 \end{pmatrix}$$

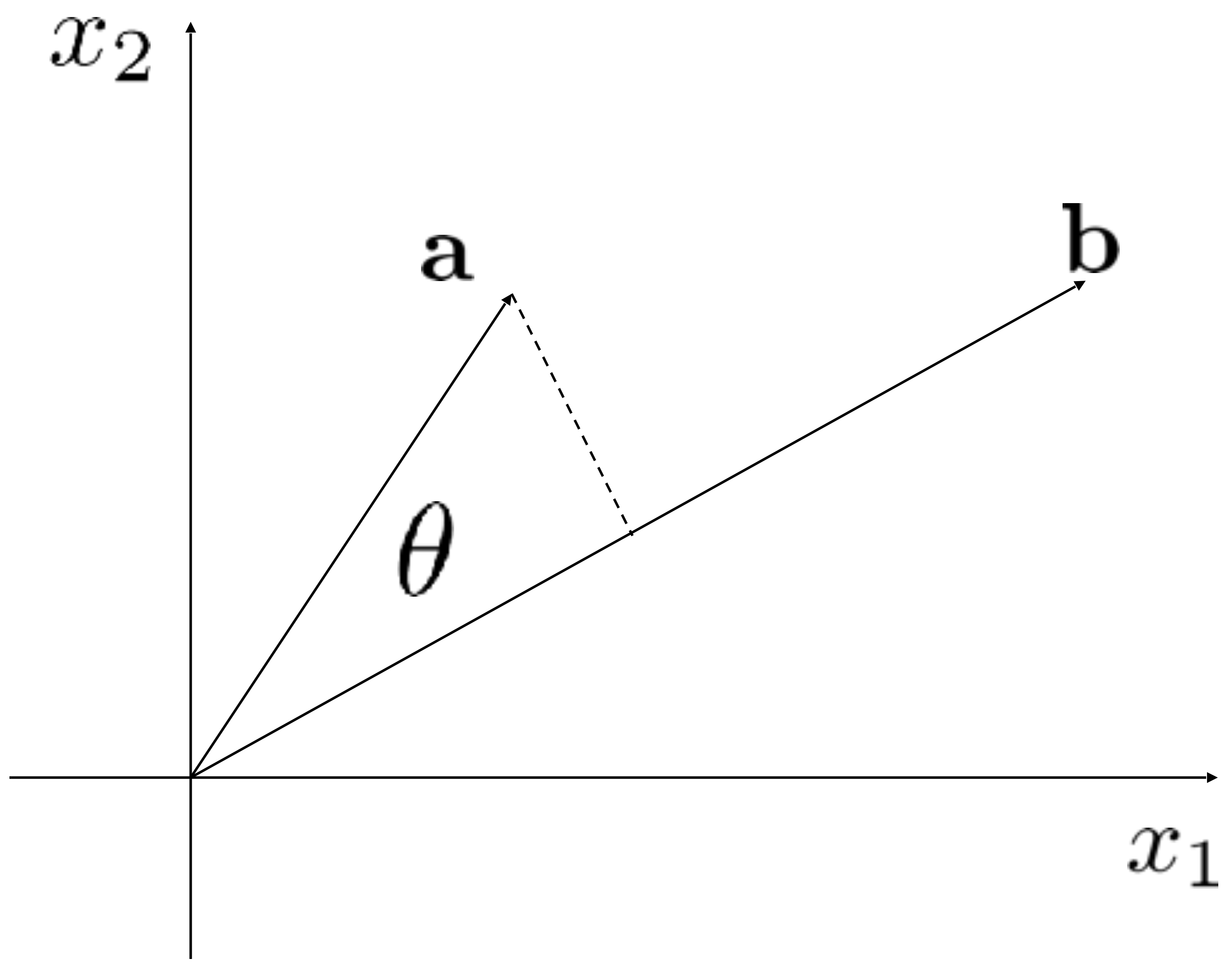
$$\text{sparrow} \cdot \text{chipmunk} = 0.01653$$

very different!

$$\text{sparrow} \cdot \text{bat} = 1.00783$$

$$\text{chipmunk} \cdot \text{bat} = 1.0171$$

Vector projection: Inner product



$$\mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \qquad \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

$$< \mathbf{a}, \mathbf{b} > \ \equiv \mathbf{a} \cdot \mathbf{b} \ \equiv \mathbf{a}^T \mathbf{b} \ \equiv a_1 b_1 + a_2 b_2 + a_3 b_3$$

It's a scalar!

$$cos(\theta) = \frac{<\mathbf{a},\mathbf{b}>}{||\mathbf{a}||_2 \times ||\mathbf{b}||_2}$$

The "cosine similarity" above can be used to measure the "similarity" between two vectors (data samples) that are not normalized (non-unit).

Cosine similarity

A cosine similarity value of 0 indicates two vectors are

A. the least similar
B. the most similar
C. the most uncertain
D. the least uncertain

Cosine similarity

	fly?	laying eggs?	weight (lb)
sparrow	yes	yes	0.087
chipmunk	no	no	0.19
bat	yes	no	0.09

Feature representation (one-hot encoded).

$$\text{sparrow} = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0.087 \end{pmatrix} \qquad \text{chipmunk} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0.19 \end{pmatrix} \qquad \text{bat} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0.09 \end{pmatrix}$$

$$\frac{\text{sparrow}\cdot\text{chipmunk}}{||\text{sparrow}||_2 \times ||\text{chipmunk}||_2} = 0.0082$$

$$\frac{\text{sparrow}\cdot\text{bat}}{||\text{sparrow}||_2 \times ||\text{bat}||_2} = 0.502$$

$$\frac{\text{chipmunk}\cdot\text{bat}}{||\text{chipmunk}||_2 \times ||\text{bat}||_2} = 0.503$$

$\cdot$ refers to the dot product between two vectors;

$||\ ||_2$ refers to the L2 norm of a vector;

$\times$ refers to the multiplication of two scalar values.

Feature scaling is another factor

Now we purposely stretch one particular feature dimension by a large factor. Let's see what will happen.

$$\text{sparrow} = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 87 \end{pmatrix} \qquad \text{chipmunk} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 190 \end{pmatrix} \qquad \text{bat} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 90 \end{pmatrix}$$

$$\frac{\text{sparrow}\cdot\text{chipmunk}}{||\text{sparrow}||_2 \times ||\text{chipmunk}||_2} = 0.99984$$

$$\frac{\text{sparrow}\cdot\text{bat}}{||\text{sparrow}||_2 \times ||\text{bat}||_2} = 0.99987$$

$$\frac{\text{chipmunk}\cdot\text{bat}}{||\text{chipmunk}||_2 \times ||\text{bat}||_2} = 0.99990$$

Now, the concept of similarity diminishes.
Conclusion: The relative scaling of the individual features is also important.

In practice, we often normalize the individual features to [0, 1] to make them directly comparable.

Cosine similarity

Interpret a dot product as the un-normalized similarity between two vectors (data samples).

The greater the dot product value is, the more similar the two data samples are. Max is 1 IFF vectors unit length

The dot product value 0 refers to the least similar two data samples, indicating two vectors that are orthogonal to each other

The cosine similarity can also be used to measure the similarity (normalized [0, 1]) between two vectors.

0 and 1 refer to the least and the most similar data samples respectively.