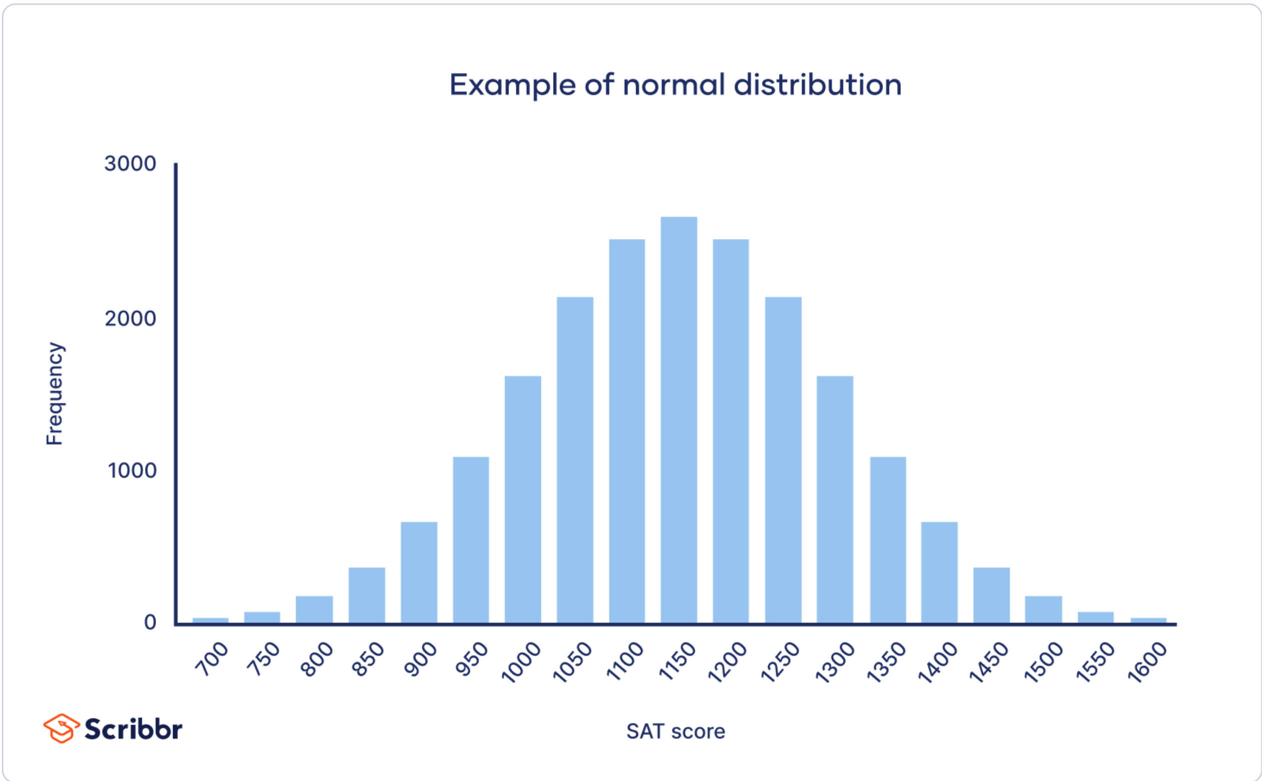
Notes: An Introduction to Vectors (MML Course)

Chapter Objective: Building motivation to learn Linear Algebra by showing how the concept of vektor, which we initially know as geometric objects, is actually crucial for solving various problems in the world of data and Machine Leaning.

---

1. The Core Problem: "Fitting" Data

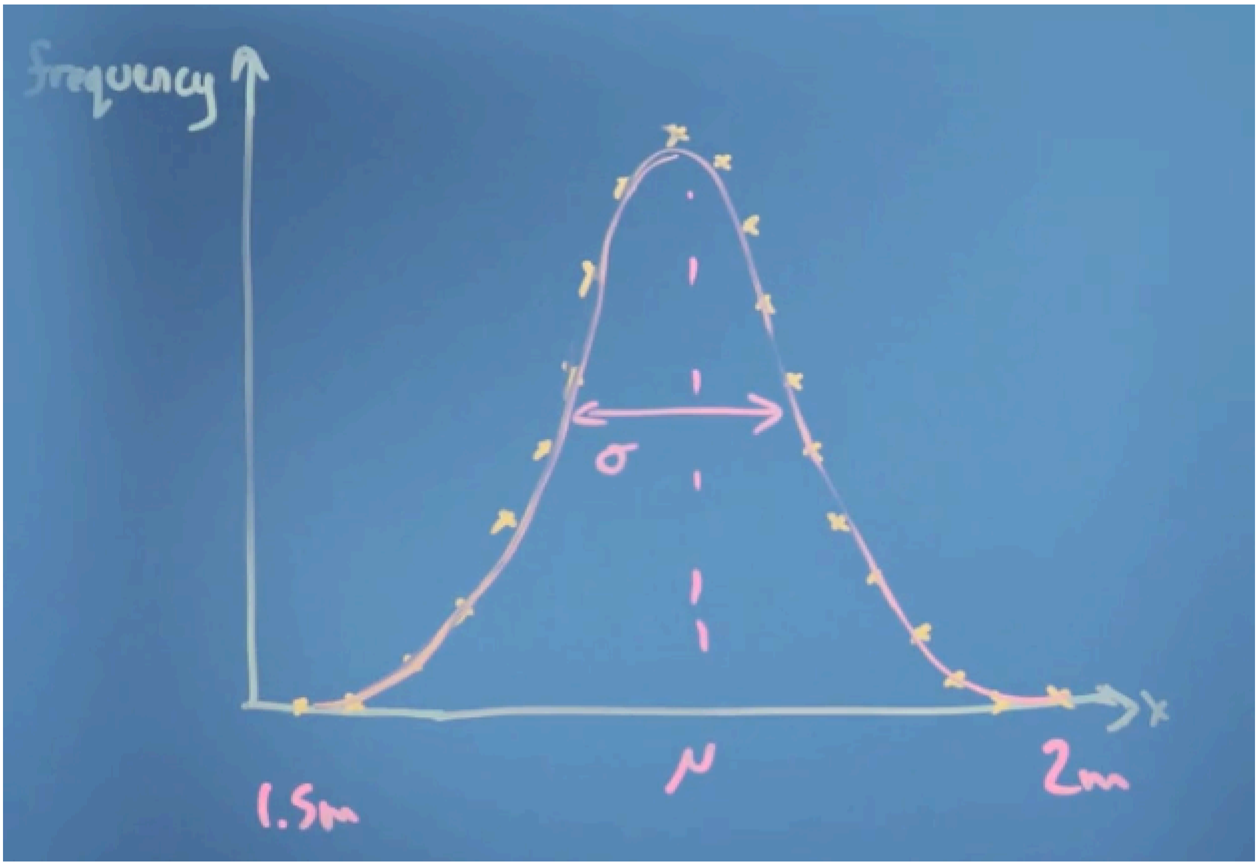
- **Initial Context**: Imagine we have a dataset representing a distribution of heights (like a histogram).



- **Goal:** Instead of just having raw data, we want to find a **mathematical function (a model)** that can ***"mimic"*** or ***"approximate"*** this data distribution
- **Example Model:** The **Normal (or Gaussian) Distribution** function
  - This function has two "knobs" or **parameters** that we can adjust:
    1. `μ` **(mu):** Controls the **center** or mean of the curve (where its peak is).
    2. `σ` **(sigma):** Controls the width or spread of the curve.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

> "Fitting" means we are **searching for the BEST values of μ and σ** that make our curve match the data histogram as closely as possible.



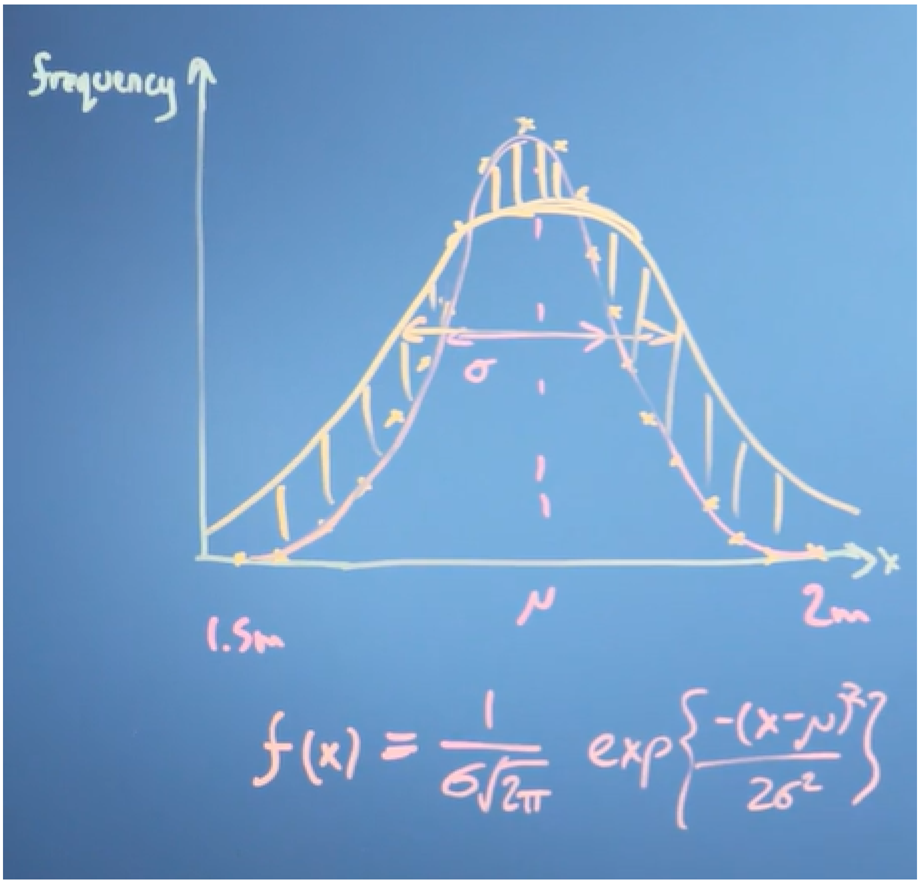
---

2. Measuring the "Badness" of a Guess

The logical next question is: "How do we know if a set of parameters is 'good' or 'bad'?" We need a "scoring metric".

How does it work?

1. **Make One Guess:** Let's say we turn our knobs to a random position: `μ = 1.8m` and `σ = 0.2m`. This produces one specific curve.
2. **Compare with the Original Data:**
   - For each bar of the histogram (e.g., at the height `1.7m`), we compare its height to the height of our curve. There will be a difference (an **error**).
   - Sometimes our curve is too high (we **overestimate**).
   - Sometimes our curve is too low (we **underestimate**).



3. **Calculate the Total "Badness Score":**
    - We cannot simply sum the differences, because the positive differences (overestimates) and negative differences (underestimates) could cancel each other out, making the score look good when it's not.
    - **Solution:** We **square each difference**. This makes all errors positive and gives a larger "penalty" to bigger errors.
    - Then, we **sum all of these squared differences**.

The final result is a single number. This number is our **"Badness Score"**.

- If the score is large, it means our guess for `μ` and `σ` was bad.
- If the score is small, it means our guess for `μ` and `σ` was good.
- If the score is zero, it means our guess was perfect.

---

3. The "Aha!" Moment: The Parameter Landscape & The Birth of Vectors

Visualizing the "Badness Score"

- Imagine a "map" where the horizontal axis is `μ` and the vertical axis is `σ`. This is called **Parameter Space**.
- For every point `(μ, σ)` on this map, we can calculate its "badness score".

- If we plot this score as "elevation", we get a 3D **landscape**, with a **valley** at the point of the best-fitting `(μ, σ)`.
- Lines of equal "elevation" form a **contour map**.

How Do We Find the Bottom of the Valley?

- We don't want to calculate the score for every possible `μ` and `σ` (it's too much work).
- **Strategy:** Start at an initial guess. Then, take a **small step** in a direction that makes our score improve (go down). Repeat this until we reach the bottom. This method called: [Gradient Descent]

This Is Where VECTORS Reappear

> A **"small step"** in this "parameter map" of `(μ, σ)` is conceptually a [Vector].

- `[change in μ, change in σ]` is a vector.
- This vector doesn't live in physical space (meters), but in a **"parameter space"**. Mathematically, however, it behaves in the same way.

---

4. An Expanded Definition of a Vector

- **Physics View (Traditional):** A vector is an arrow in physical space (position, velocity). [Ch 01 - Vectors.md]
- **Computer Science View (Modern):** A vector is simply an ordered **LIST OF NUMBERS**.
  - **Car Example:** `[price, CO2 emissions, safety rating, top speed]` is a vector describing a car.
  - **Alloy Example:** `[percentage of Iron, percentage of Carbon, ...]` is a vector describing an alloy.
  - **Einstein's Spacetime:** `[x, y, z, t]` is a 4-dimensional vector.
- **Conclusion:** Thinking of a "move" in the `(μ, σ)` parameter space as a vector is a very natural and powerful thing to do.

---

5. The Course Roadmap

- **Ultimate Goal:** Find the bottom of the valley in the parameter landscape (perform **optimization**).
- **Tools We Need:**
  1. **Linear Algebra (Vectors):** To understand how to "move" in parameter space.
  2. **Calculus (Turunan):** To figure out which direction is the steepest way down in that landscape (to find the **gradient**).

By combining these two tools, we can perform optimization, which is the engine that enables us to do Machine Learning.

Tags: #mml-specialization #linear-algebra #vectors #parameter-space #optimization