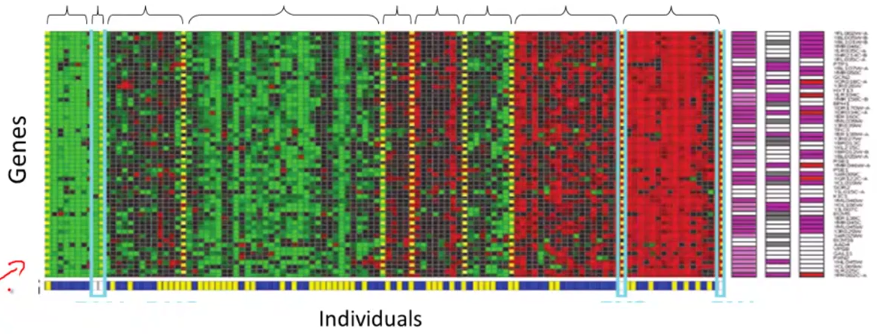
* **Supervised Learning 🡪** given a data set and *already know* what our correct output (“right answer”) should look like, while also having the idea that there is a relationship between the input and the output.
* Supervised learning problems are categorized into **regression** and **classification** problems.
* regression 🡪 trying to predict results within a *continuous* output, or we are trying to *map input variables to some continuous function*.
* Classification 🡪 trying to predict results in a *discrete* output (0 or 1), or we are trying to map *input variables into discrete categories*.
* Example 1: Given data about the size of houses on the real estate market, try to predict their price 🡪 Price = a function of size 🡪 a continuous output, so this is a regression problem.
* We could turn this example into a classification problem by instead making our output = *whether a house sells for more than or less than the asking price.*
* Here we are classifying the houses based on price into 2 discrete categories (> or <)
* Example 2:
* (a) Regression - Given a picture of a person, predict age based on the given picture
* (b) Classification - Given a patient w/ a tumor, predict whether it is malignant or benign.
* **Unsupervised learning** allows us to approach problems w/ little/no idea what results should look like 🡪 given a data set w/ no labels or what each it, find some structure w/in it
* can derive structure from data where we don't necessarily know the effect of the variables.
* can derive this structure by **clustering** data based on relationships among variables in the data.
* NO feedback based on the prediction results.
* Example 1: Clustering:
* Take a collection of 1,000,000 different genes and find a way to automatically group them into *groups* that are *somehow similar*/related by different variables, such as lifespan, location, roles, and so on.
* Take a collection of genes being expressed in individuals (via color gradient) and group individuals into similar groups without knowing types of people in advance (no right answer)



* Google News 🡪 looks throughout stories on the web and groups them into cohesive news stories (multiple URL’s in each cluster to display together)
* Organize large CPU clusters to see which machines work better together
* Social Network analysis 🡪 by knowing who you email/Facebook friends, etc. can we automatically ID cohesive groups of friends that all know each other
* Market Segmentation 🡪 look at customer data set and group customers into different market segments to more efficiently market to different segments
* Astronomical Data Analysis 🡪 useful theories on how galaxies are formed
* Example 1: Non-clustering:
* The "**Cocktail Party Algorithm**", allows you to find structure in a chaotic environment. (i.e. identifying individual voices and music from a mesh of sounds at a [cocktail party](https://en.wikipedia.org/wiki/Cocktail_party_effect)).

1. 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**. Suppose we feed a learning algorithm a lot of historical weather data, and have it learn to predict weather. What would be a reasonable choice for P?

* **The probability of it correctly predicting a future date's weather.**

1. The amount of rain that falls in a day is usually measured in either millimeters (mm) or inches. Suppose you use a learning algorithm to predict how much rain will fall tomorrow. Would you treat this as a classification or a regression problem?

* **Regression**

1. Suppose you are working on stock market prediction, and you would like to predict the price of a particular stock tomorrow (measured in dollars). You want to use a learning algorithm for this. Would you treat this as a classification or a regression problem?

* **Regression**

1. Some of the problems below are best addressed using a supervised learning algorithm (*we teach CPU*), and the others with an unsupervised learning algorithm (*CPU teaches itself)*. In each case, assume some appropriate dataset is available for your algorithm to learn from.

* Given data on crop yields over the last 50 years, learn to predict next year's yields 🡪 **Supervised**
* Given data on how 1000 patients respond to an experimental drug (effectiveness of treatment, side effects, etc.), discover whether there are different categories or "types" of patients in terms of how they respond to the drug, and if so what these categories are. 🡪 **Unsupervised**
* Examine a web page, and classify whether the content on the page should be considered "child friendly" (e.g., non-pornographic, etc.) or "adult." 🡪 **Supervised**
* Given a large dataset of medical records from patients suffering from heart disease, try to learn whether there might be different clusters of such patients for which we might tailor separate treatments. 🡪 **Unsupervised**

1. Which of these is a reasonable definition of machine learning?

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

1. An email program watches which emails we mark as spam or not spam, and based on this activity learns how to better filter spam. What is the task, T, here?

* **Classify emails as spam or not spam**
* **E = watching you classify spam and not spam**
* **P = probability the program will correctly ID /not spam, OR the # or fraction/% of emails correctly classified as spam/not spam**