

Day 2: Linear Regression

Summer STEM: Machine Learning

Department of Electrical and Computer Engineering
NYU Tandon School of Engineering
Brooklyn, New York

Outline

- 1** Leftovers from Day 1
- 2** Introduction to Machine Learning
- 3** Statistics Basics
- 4** Linear Regression

Exercises: Matrix Multiplication

- $X = \begin{bmatrix} 1 & 2 & -1 \\ 1 & 0 & 1 \end{bmatrix}$ $Y = \begin{bmatrix} 3 & 1 \\ 0 & -1 \\ -2 & 3 \end{bmatrix}$ $Z = \begin{bmatrix} 1 \\ 4 \\ 6 \end{bmatrix}$
- Calculate XY , YX , $Z^T Y$

Matrix Inverse

- Analogy: Reciprocal of a number $\frac{1}{a}a = 1$
- Matrix inverse only defined for square matrix (# rows = # cols)
- $A^{-1}A = AA^{-1} = I$. I is called the identity matrix, whose diagonal elements are 1 and other elements are 0.
- Hard to compute by hand, but for 2 by 2 matrix, it is

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

- Even for a square matrix, the matrix inverse does not always exist. Can you tell when that is the case for 2 by 2 matrices based on the formula given above?

Matrix Inverse

When is matrix inverse useful? We can use it to solve systems of linear equations!

- Consider the following equations

$$\begin{cases} x + 2y = 5 \\ 3x + 5y = 13 \end{cases}$$

- In matrix-vector form

$$\begin{bmatrix} 1 & 2 \\ 3 & 5 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 5 \\ 13 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 \\ 3 & 5 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 2 \\ 3 & 5 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 3 & 5 \end{bmatrix}^{-1} \begin{bmatrix} 5 \\ 13 \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 3 & 5 \end{bmatrix}^{-1} \begin{bmatrix} 5 \\ 13 \end{bmatrix}$$

Demo and Exercises: NumPy

Open `demo_vectors_matrices.ipynb`

- Your task: use NumPy functions to compute the exercises we did earlier this morning. Compare the results.

Demo: Plotting Functions

- Generate and plot the following functions in Python:
 - Scatter plot of points: (0,1), (2,3), (5,2), (4,1)
 - Straight Line: $y = mx + b$
 - Sine-wave $y = \sin(x)$
 - Polynomial e.g. $y = x^3 + 2$
 - Exponential e.g. $y = e^{-2x}$
 - Choose a function of your own
- Use Wikipedia and Numpy Documentation to search for mathematical formulas and python functions

Looking at our ice-breaker data in spreadsheets

- Columns have labels in the first row
- Collected data (numbers, words) follow below
- Let's export it to a Comma-Separated Values (CSV) file and open it

Outline

1 Leftovers from Day 1

2 Introduction to Machine Learning

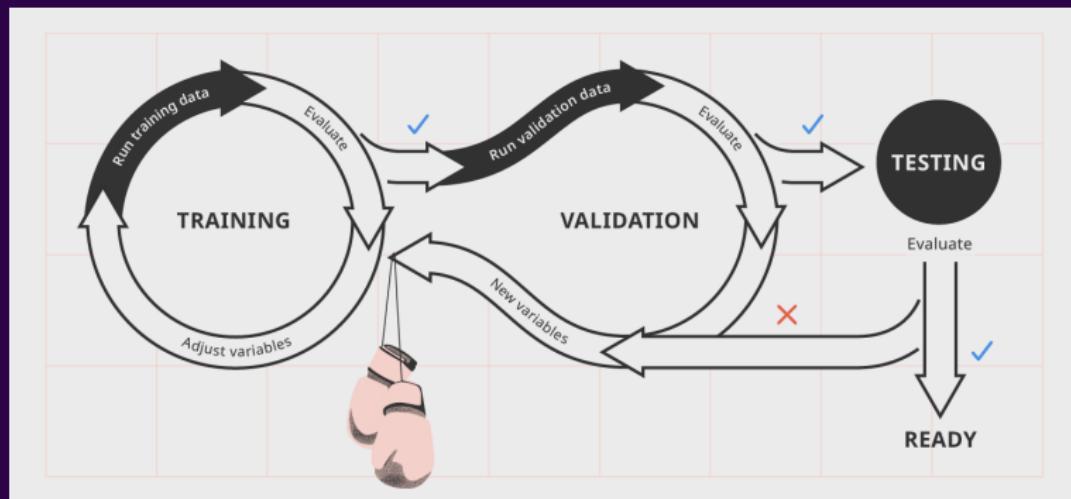
3 Statistics Basics

4 Linear Regression

What is Machine Learning

- Recognize patterns from data
- Make predictions based on the learnt patterns
- A very effective tool where human expertise is not available

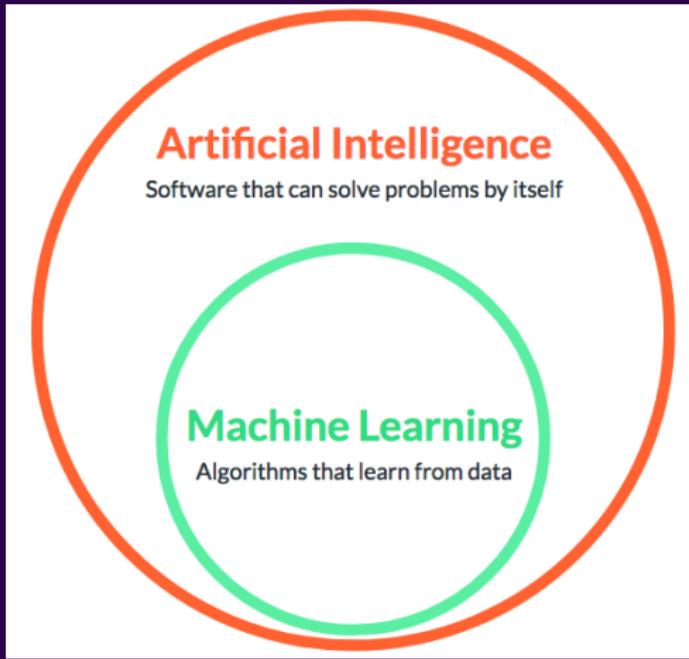
Machine Learning Pipeline



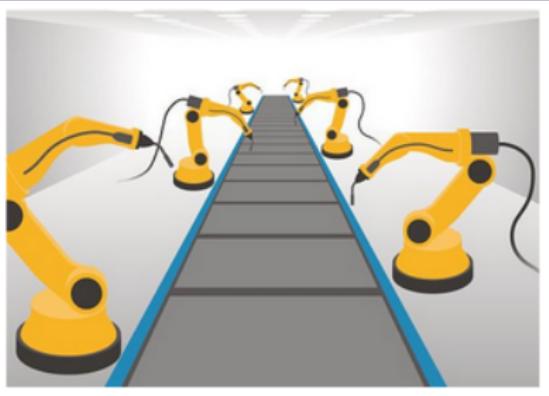
Artificial Intelligence

- Search
- Reasoning and Problem Solving
- Knowledge Representation
- Planning
- Learning
- Perception
- Natural Language Processing
- Motion and Manipulation
- Social and General Intelligence

Machine Learning



Autonomous vs. Automated



Autonomous Example: Self-driving car



- Waymo Video

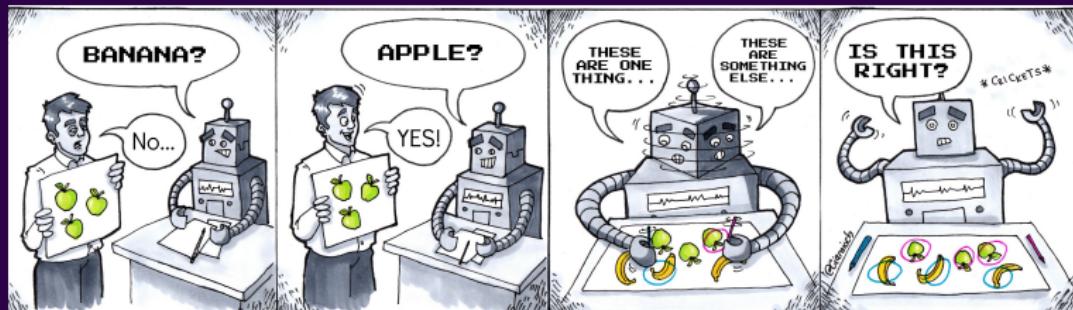
Why is Machine Learning so Prevalent?

- Database mining
- Medical records
- Computational biology
- Engineering
- Recommendation systems
- Understanding the human brain

Why Now?

- Big Data
 - Massive storage. Large data centers
 - Massive connectivity
 - Sources of data from internet and elsewhere
- Computational advances
 - Distributed machines, clusters
 - GPUs and hardware

Supervised Vs. Unsupervised Learning



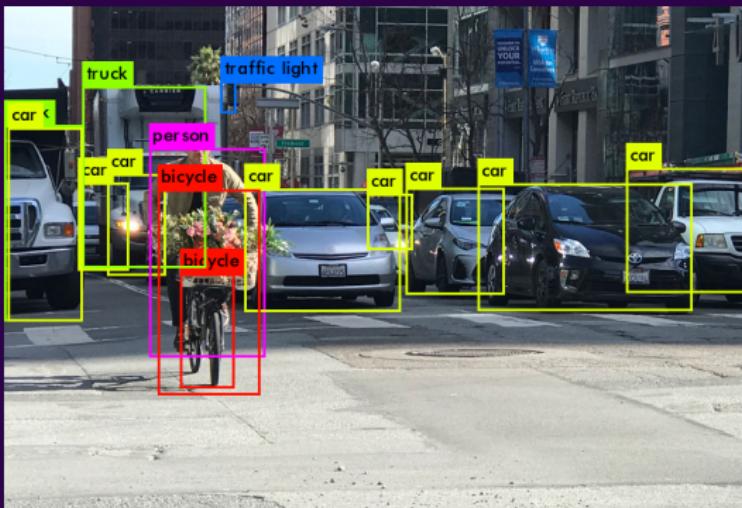
Supervised Learning

Unsupervised Learning

Supervised Vs. Unsupervised Learning

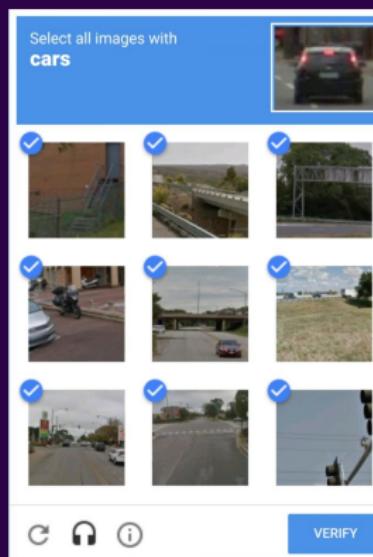
- The main difference between supervised and unsupervised learning is the existence of a supervisor, which in many cases is in the form of a data label.
- The label of the data is what we want the machine learning algorithm to predict.

Labelled Data

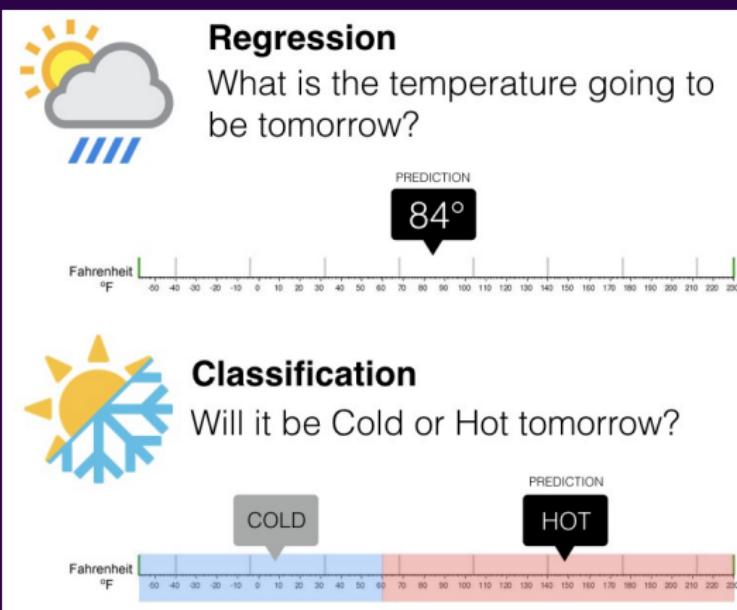


■ YOLO v2

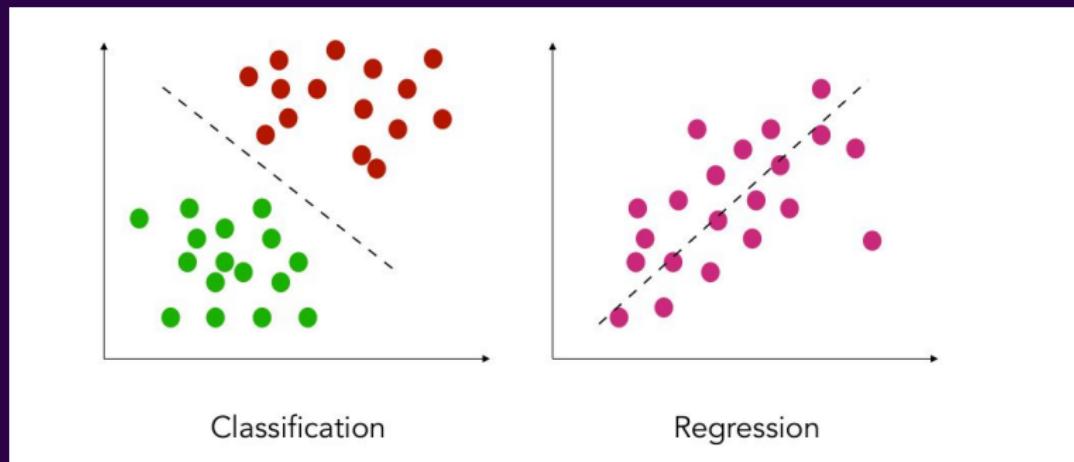
How labels are generated



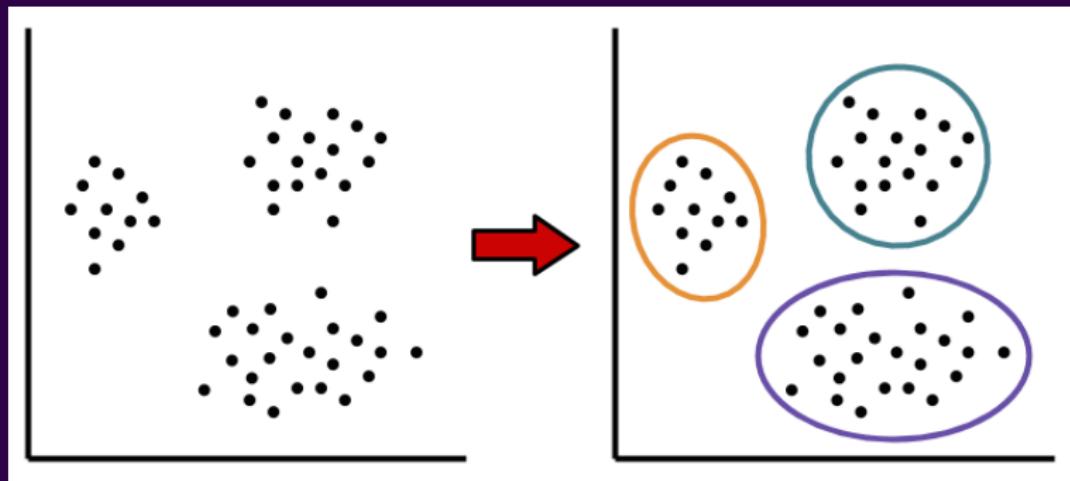
Classification Vs. Regression



Classification Vs. Regression



Unsupervised Learning



source: the dish on science

Outline

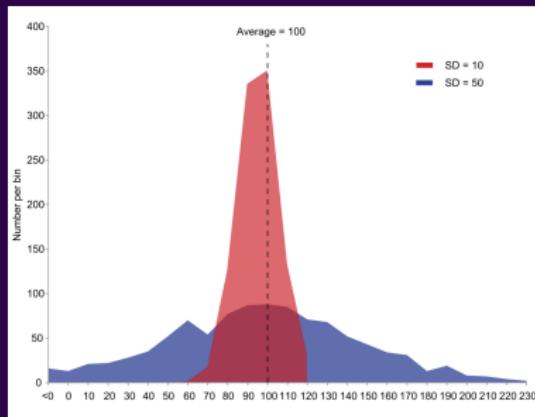
- 1** Leftovers from Day 1
- 2** Introduction to Machine Learning
- 3** Statistics Basics
- 4** Linear Regression

Basic Concepts

- **Mean** (average value): $\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i$
- **Variance** describes the spread of the data with respect to the mean.
- **Covariance** describes the relationship between two variables.

Variance

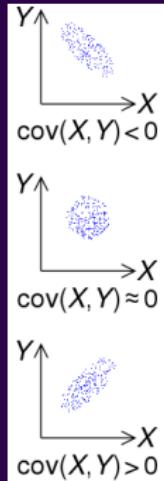
$$\text{■ Variance: } \sigma_x^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2$$



<https://en.wikipedia.org/wiki/Variance>

Covariance

- Covariance: $\sigma_{xy} = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y})$



Outline

- 1** Leftovers from Day 1
- 2** Introduction to Machine Learning
- 3** Statistics Basics
- 4** Linear Regression

Supervised learning in a nutshell

Given the dataset (x_i, y_i) for $i = 1, 2, \dots, N$, find a function $f(x)$ (model) so that it can predict the label \hat{y} for some input \hat{x} , even if it is not in the dataset, i.e. $\hat{y} = f(\hat{x})$.

Many possible models

- $f(x) = w_1 x + w_0$.
- $f(x) = w_2 x^2 + w_1 x + w_0$.
- $f(x) = \frac{1}{e^{-(w_1 x + w_0)} + 1}$.
- The numbers w_0 , w_1 and w_2 are called model parameters.
- We often write the model as $f(x; \mathbf{w})$, stacking all parameters to a vector \mathbf{w} .

How would you fit a line?

Can you find a line that passes through $(0, 0)$ and $(1, 1)$?

How would you fit a quadratic curve?

Can you find a quadratic curve that passes through $(0, 0)$, $(1, 1)$ and $(-1, 1)$?

What model do we use for this dataset?

- Open `demo_boston_housing_1d.ipynb`
- Can you find a line that go through ALL of the data points?
Why?

Is Your Model a Good Fit?

- How would you determine if your model is a good fit or not?
 - How will you determine this?
 - Is there a quantitative way?

Error Functions

- An **error function** quantifies the discrepancy between your model and the data.
 - They are non-negative, and go to zero as the model gets better.
- Common Error Functions:
 - Mean Squared Error: $MSE = \frac{1}{N} \sum_{i=1}^N \|y_i - \hat{y}_i\|^2$
 - Mean Absolute Error: $MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|$
- In later units, we will refer to these as **cost functions** or **loss functions**.
- Compute MSE on your model
- How do we interpret MSE? MAE?

Least Square Solution

- Model:

$$f(x) = w_0 + w_1 x$$

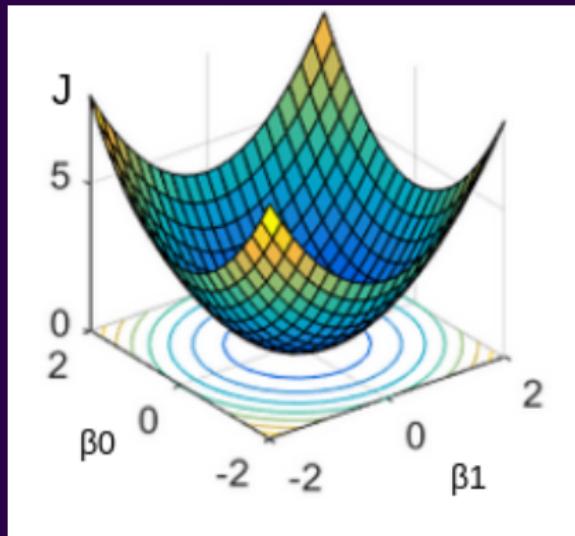
- Loss:

$$J(w_0, w_1) = \frac{1}{N} \sum_{i=1}^N \|y_i - f(x_i)\|^2$$

- Optimization: find w_0, w_1 such that $J(w_0, w_1)$ is the least possible value (hence the name “least square”).

Loss Landscape

Plot the loss against the parameters:



Least Square Solution

■ Solution:

$$f(x) = \bar{y} + \frac{\sigma_{xy}}{\sigma_x^2}(x - \bar{x})$$

$$w_1 = \frac{\sigma_{xy}}{\sigma_x^2}, \quad w_0 = \bar{y} - w_1 \bar{x}$$

■ Prediction:

$$\hat{y} = w_0 + w_1 \hat{x}$$

What model do we use for this dataset?



<https://www.explainxkcd.com>

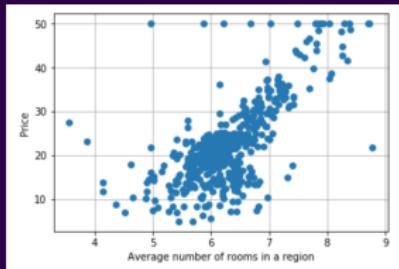
Exercise

- Open `demo_boston_housing_1d.ipynb`
- Compute the least square solution in NumPy. Compare the results with the ones from sklearn.

Lab: Statistics

■ Data Representation:

- y = variable you are trying to predict. Also referred to as: label, dependent variable, response variable, target variable etc.
- x = what you are using to predict. Also referred to as: feature, independent variable, attribute, predictor etc.
- Set of points, (x_i, y_i) , $i = 1, \dots, N$. Each data point is called a sample.
- An efficient way to visualize the data is by plotting x vs y in a scatter plot.



Linear Regression

- Today: linear models! In 1D, this is $f(x) = w_1x + w_0$
- One of the simplest machine learning model, yet very powerful.
- Rewrite the set of linear equations into matrix form.

Linear Regression

- For N data points (x_i, y_i) we have,

$$y_1 \approx w_0 + w_1 x_1$$

$$y_2 \approx w_0 + w_1 x_2$$

$$\vdots$$

$$y_N \approx w_0 + w_1 x_N.$$

Linear Regression

- In matrix form we have,

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \approx \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_N \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}$$

- We can write it as $Y \approx X\mathbf{w}$. We call X the design matrix.
- Exercise: verify $\|Y - X\mathbf{w}\|^2 = \sum_{i=1}^N \|y_i - (w_0 + w_1 x_i)\|^2$

Linear Least Square

- $\min_{\mathbf{w}} \frac{1}{N} \|Y - X\mathbf{w}\|^2$
- Using the psuedo-inverse (only square matrices have an inverse),

$$Y = X\mathbf{w}$$

$$X^T Y = X^T X\mathbf{w}$$

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

$$(X^T X)^{-1} X^T Y = \mathbf{w}.$$

- Exercise: open `demo_boston_housing_1d.ipynb`. Use the formula above and compare the results.

Linear Regression

- What if we have multivariate data with \mathbf{x} being a vector?
- $y = \mathbf{w}^T \mathbf{x}$, here both \mathbf{w} and \mathbf{x} are vectors.
- Ex: $\mathbf{x}_i = [1, x_{i1}, x_{i2}]^T$ and $\mathbf{w} = [w_0, w_1, w_2]^T$
 $y_1 \approx w_0 + w_1 x_{11} + w_2 x_{12}, \dots$
- In matrix-vector form

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \approx \begin{bmatrix} 1 & x_{11} & x_{12} \\ 1 & x_{21} & x_{22} \\ \vdots & \vdots & \vdots \\ 1 & x_n & x_{n2} \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix}$$

- Solution remains the same $(X^T X)^{-1} X^T Y = \mathbf{w}$
- Exercise: open `demo_multilinear.ipynb`

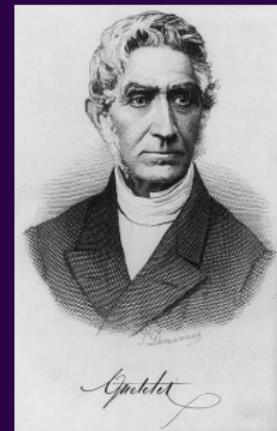
This the best linear fit according to...



Carl Friedrich Gauss



Adrien-Marie Legendre



Adolphe Quetelet