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 Leaning

3 Lab: Simple Linear Model

4 Lab: Goodness of Fit

5 Statistics Basics

6 Least Squares Solution

Exercises: Matrix Multiplication

$$(AB)_{ij} = \sum_{k=1}^{N} A_{ik} B_{kj} \qquad (A^T)_{ij} = A_{ji}$$

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

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

$$Z^T = \begin{bmatrix} 1 & 4 & 6 \end{bmatrix}$$

$$Z^T Y = \begin{bmatrix} 1 & 4 & 6 \end{bmatrix} \begin{bmatrix} 3 & 1 \\ 0 & -1 \\ -2 & 3 \end{bmatrix} = \begin{bmatrix} 3 + (-12) & 1 - 4 + 18 \end{bmatrix} = \begin{bmatrix} -9 & 15 \end{bmatrix}$$

Matrix Inverse

A m by n matrix

$A^{-1}A = AA^{-1}$ m has to be equal to n

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

Ex: 2 by 2 identity matrix $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

$$[a]^{-1} = \left[\frac{1}{a}\right] \qquad \begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \underbrace{\frac{1}{ad - bc}} \underbrace{\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}}$$

- 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? $ad - bc = 0$

NYU TANDON SCHOOL OF ENGINEERING

Matrix Inverse

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

- Consider the following equations

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

$$Iu = u$$

$$\text{Ex } \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} u_1 + 0 \\ 0 + u_2 \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

- In matrix-vector form

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

$$\begin{bmatrix} 1 & 2 \\ 3 & 5 \end{bmatrix}^{-1}$$

$$= \frac{1}{5 \times 1 - 2 \times 3} \begin{bmatrix} 5 & -2 \\ -3 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} -5 & 2 \\ 3 & -1 \end{bmatrix}$$

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

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

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 Leaning

3 Lab: Simple Linear Model

4 Lab: Goodness of Fit

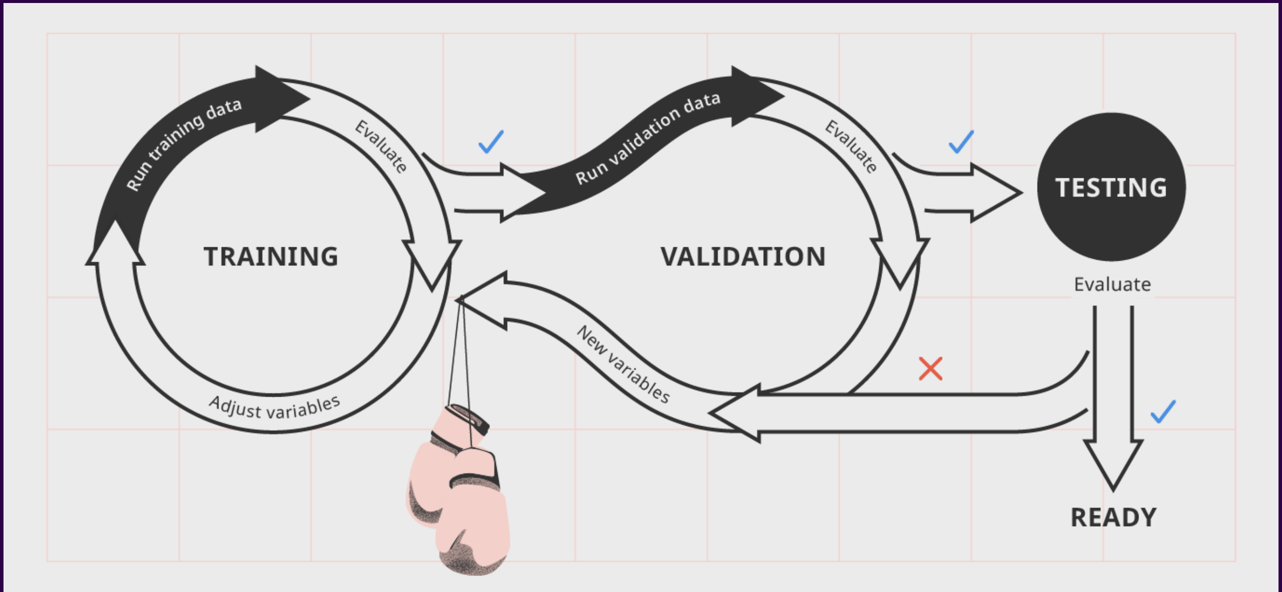
5 Statistics Basics

6 Least Squares Solution

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

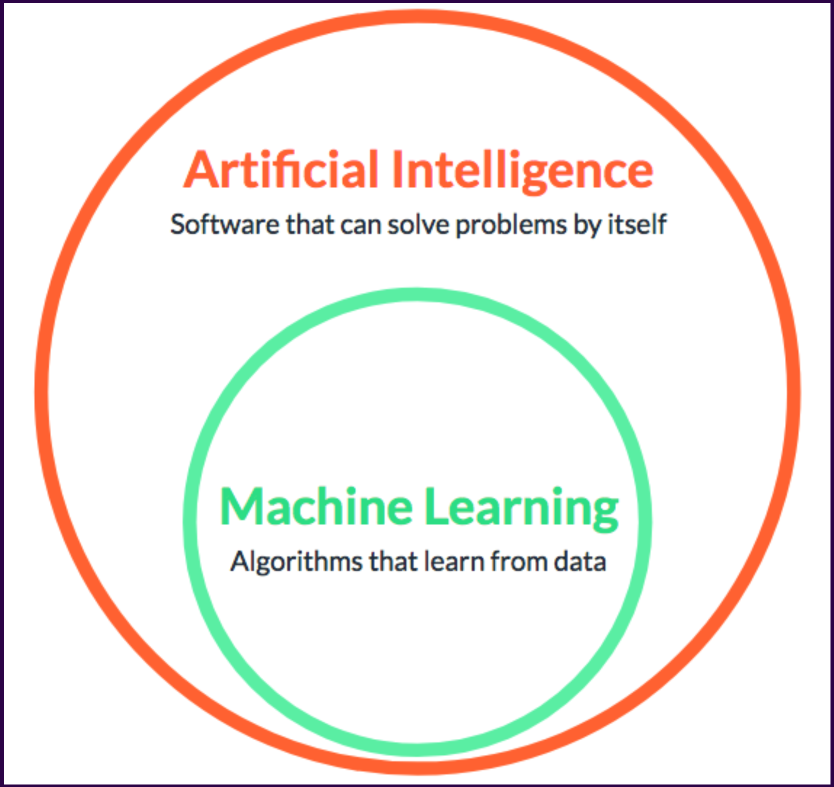


https://lionbridge.ai/articles/how-does-ai-training-work/

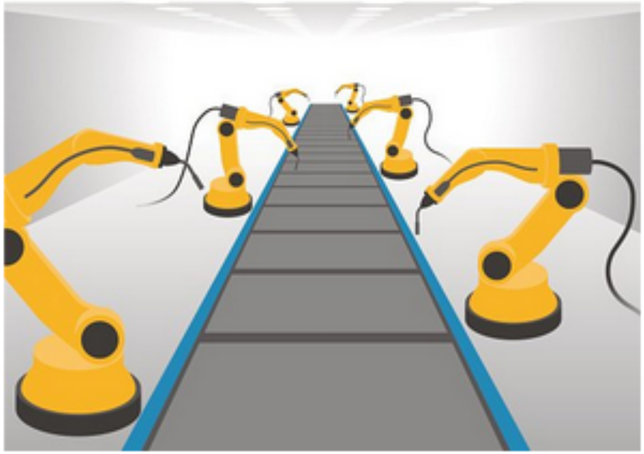
Artificial Intelligence

- 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

https://www.tesmanian.com/blogs/tesmanian-blog/tesla-autopilot-full-self-driving-fsd-improvements-video

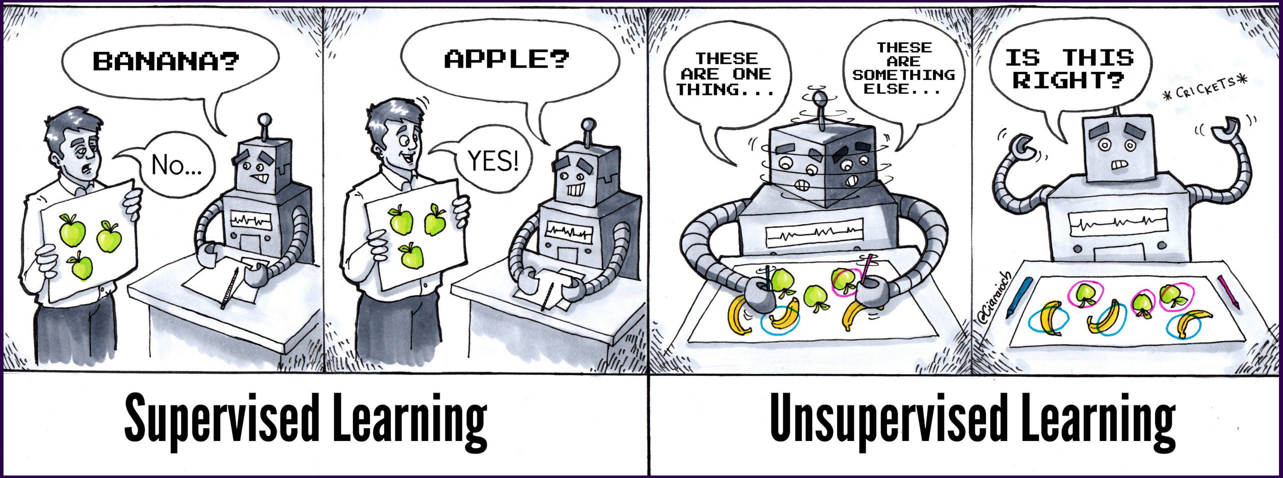
Why is Machine Learning so Prevalent?

- 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

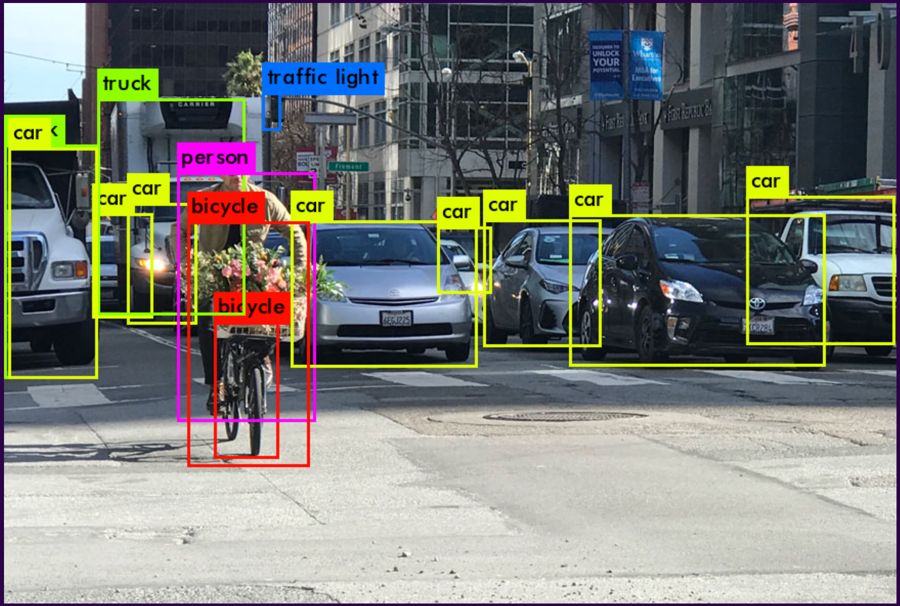


twitter.com/athena_schools/status/1063013435779223553/photo/1

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.

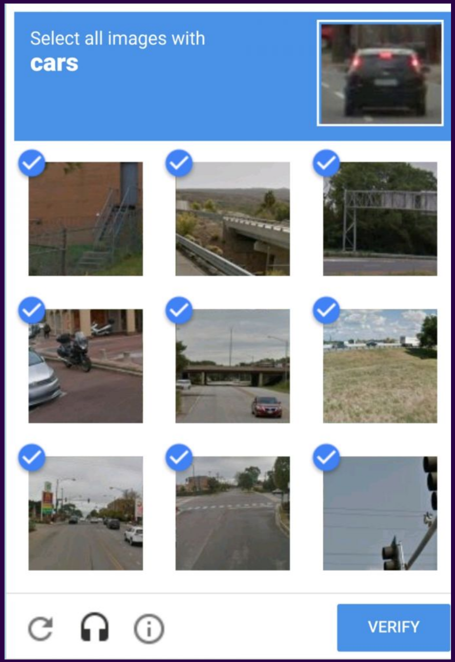
Labelled Data



- YOLO v2

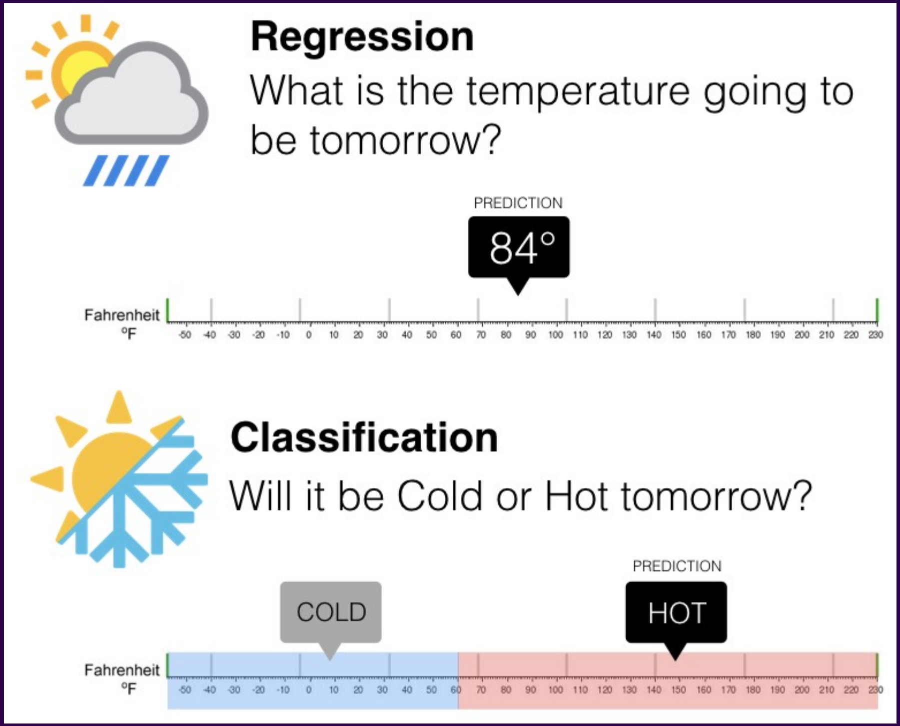
https://towardsdatascience.com/yolo-you-only-look-once-17f9280a47b0

How labels are generated



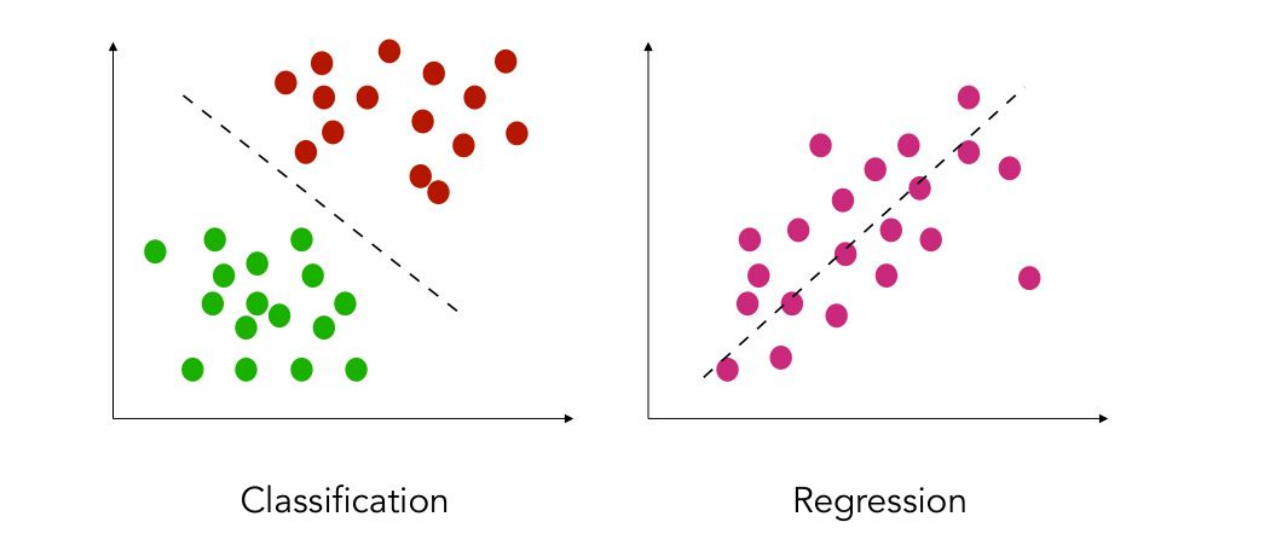
https://devrant.com/rants/1758134/select-all-images-with-cars-i-did-and-its-not-correct-why-not

Classification Vs. Regression



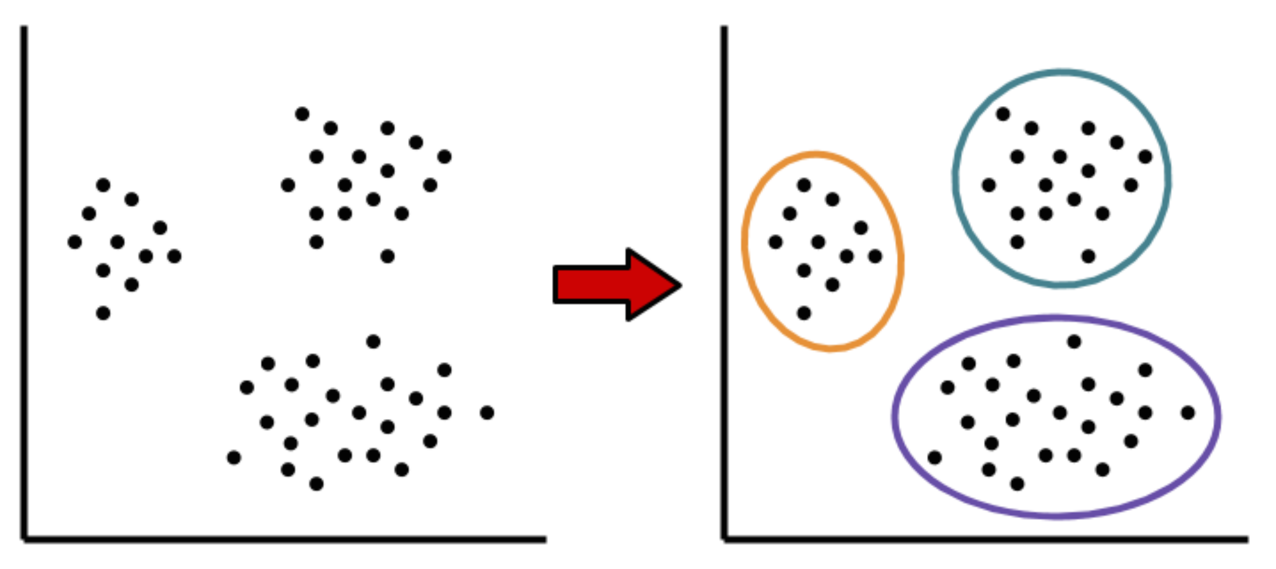
https://www.pinterest.com/pin/672232681855858622/?lp=true

Classification Vs. Regression



https://maelfabien.github.io/machinelearning/ml_base/

Unsupervised Learning



source: the dish on science

Outline

1 Leftovers from Day 1

2 Introduction to Machine Leaning

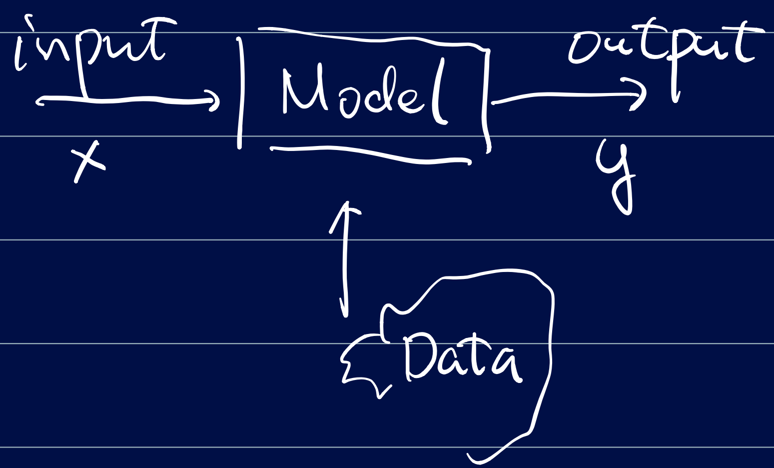
3 Lab: Simple Linear Model

4 Lab: Goodness of Fit

5 Statistics Basics

6 Least Squares Solution

Machine Learning



Ex: x : movie genre

y : IMDB score

Dataset: $\underbrace{(x_i, y_i)}_{\text{a sample}}$ $i = 1, 2, \ldots N$

We also call x features, y labels

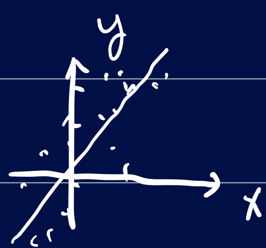
Task: to find a model $f(x)$ to predict

$$\hat{y} = f(x)$$

($\hat{y}$: "y hat", denotes the prediction given by the model)

Ex : $\hat{y} = f(x) = \underset{\sim}{w_1} x + \underset{\sim}{w_0}$

" linear model "



$w_0 = 0$

$w_1 = 2$

w_0, w_1 : parameters of the model

Cost functions : measure how well the model fits the data

MSE : $\frac{1}{N} \sum_{i=1}^{N} \| y_i - \hat{y}_i \|^2$

y_i: the labels in the dataset
"groud truth" label

$\hat{y}_i = f(x_i)$ predictions

Compute MSE in NumPy

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \quad \hat{y} = \begin{bmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_N \end{bmatrix} \quad \text{MSE}: \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

$$y - \hat{y} = \begin{bmatrix} y_1 - \hat{y}_1 \\ y_2 - \hat{y}_2 \\ \vdots \\ y_N - \hat{y}_N \end{bmatrix}$$

In NumPy, $z = (y - \hat{y}) ** 2$, this will square every element $\begin{bmatrix} (y_1 - \hat{y}_1)^2 \\ \vdots \\ (y_N - \hat{y}_N)^2 \end{bmatrix}$

In our exercise $N = 506$

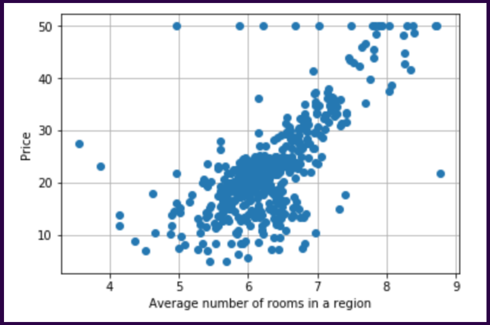
np.mean(z) will give $\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$

(For a vector z with N elements

np.mean(z) gives $\frac{1}{N}\sum_{i=1}^{N} z_i$)

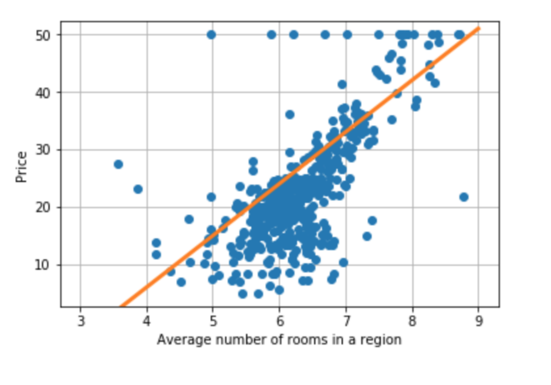
Linear Model

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



Linear Model

- Assume a linear relationship $y = b + wx$
  - $b =$ intercept
  - $w =$ slope
- $\mathbf{w} = (b, w) = (w_0, w_1)$ are the parameters of the model



- Let's go to the lab to understand this further.

Outline

1 Leftovers from Day 1

2 Introduction to Machine Leaning

3 Lab: Simple Linear Model

4 Lab: Goodness of Fit

5 Statistics Basics

6 Least Squares Solution

Is Your Model a Good Fit?

- How would you determine if your model is a good fit or not?
- Talk with your classmates next to you to see whose model fits the data the best
  - How will you determine this?
  - Is there a quantitative way?
  - Write python code if so.

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?
  - RMSE?

General Steps to Solve a Machine Learning Problem

- Load and visualize data

General Steps to Solve a Machine Learning Problem

- Load and visualize data
  - (x_i, y_i), $i = 1, ..., n$

General Steps to Solve a Machine Learning Problem

- Load and visualize data
  - (x_i, y_i), $i = 1, ..., n$
- Find an appropriate model to fit the data

General Steps to Solve a Machine Learning Problem

- Load and visualize data
  - (x_i, y_i), $i = 1, \ldots, n$
- Find an appropriate model to fit the data
  - Eg: Linear model is $\hat{y} = wx + b$

General Steps to Solve a Machine Learning Problem

- Load and visualize data
  - (x_i, y_i), $i = 1, \ldots, n$
- Find an appropriate model to fit the data
  - Eg: Linear model is $\hat{y} = wx + b$
- Choose an appropriate error function

General Steps to Solve a Machine Learning Problem

- Load and visualize data
  - (x_i, y_i), $i = 1, ..., n$
- Find an appropriate model to fit the data
  - Eg: Linear model is $\hat{y} = wx + b$
- Choose an appropriate error function
  - $MSE = \sum_{i=1}^{N}(y_i - (b + wx_i))^2$

NYU TANDON SCHOOL OF ENGINEERING

General Steps to Solve a Machine Learning Problem

- Load and visualize data
  - (x_i, y_i), $i = 1, ..., n$
- Find an appropriate model to fit the data
  - Eg: Linear model is $\hat{y} = wx + b$
- Choose an appropriate error function
  - $MSE = \sum_{i=1}^{N}(y_i - (b + wx_i))^2$
- Find parameters that minimize the error function

NYU TANDON SCHOOL OF ENGINEERING

General Steps to Solve a Machine Learning Problem

- Load and visualize data
  - (x_i, y_i), $i = 1, \ldots, n$
- Find an appropriate model to fit the data
  - Eg: Linear model is $\hat{y} = wx + b$
- Choose an appropriate error function
  - $MSE = \sum_{i=1}^{N}(y_i - (b + wx_i))^2$
- Find parameters that minimize the error function
  - Select b, w to minimize the error function

NYU TANDON SCHOOL OF ENGINEERING

Least Squares Fit

- The **Least Squares Fit** is characterized by the minimization of the MSE error function:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

Least Squares Fit

- The **Least Squares Fit** is characterized by the minimization of the MSE error function:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

  - Find the parameters, $\mathbf{w} = (b, w) = (w_0, w_1)$, that give the smallest MSE

Least Squares Fit

- The **Least Squares Fit** is characterized by the minimization of the MSE error function:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

  - Find the parameters, $\mathbf{w} = (b, w) = (w_0, w_1)$, that give the smallest MSE
- MSE is a useful metric because there exists an analytic solution to find the optimal parameters b and w

Outline

1 Leftovers from Day 1

2 Introduction to Machine Leaning

3 Lab: Simple Linear Model

4 Lab: Goodness of Fit

5 Statistics Basics

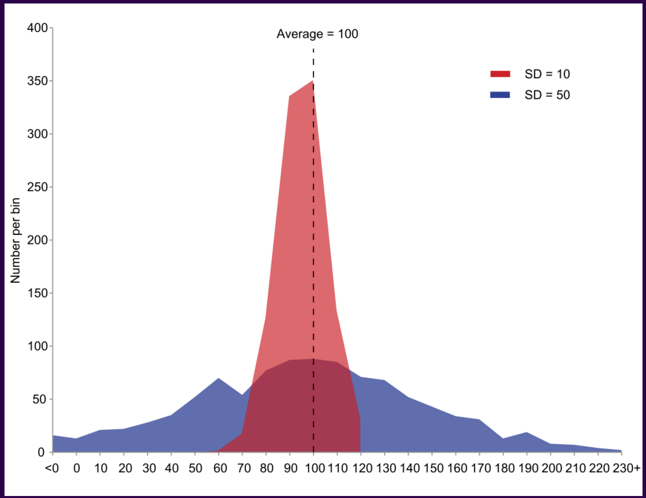
6 Least Squares Solution

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

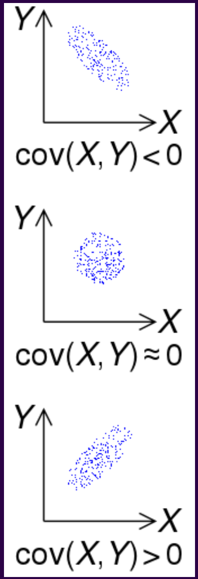
- Variance: $\sigma_x^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2$



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

Covariance

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



https://en.wikipedia.org/wiki/Covariance

Mean, Variance, and Covariance, Correlation Coefficient

- Given feature-target data
  (x_i, y_i), $i = 1, 2, ..., N$

Mean, Variance, and Covariance, Correlation Coefficient

- Given feature-target data (x_i, y_i), $i = 1, 2, ..., N$
- Mean:

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i, \quad \bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i$$

Mean, Variance, and Covariance, Correlation Coefficient

- Given feature-target data (x_i, y_i), $i = 1, 2, ..., N$
- Mean:

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i, \quad \bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i$$

- Variance:

$$\sigma_x^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2, \quad \sigma_y^2 = \frac{1}{N}\sum_{i=1}^{N}(y_i - \bar{y})^2$$

Mean, Variance, and Covariance, Correlation Coefficient

- Given feature-target data (x_i, y_i), $i = 1, 2, ..., N$
- Mean:

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i, \quad \bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i$$

- Variance:

$$\sigma_x^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2, \quad \sigma_y^2 = \frac{1}{N}\sum_{i=1}^{N}(y_i - \bar{y})^2$$

- **Covariance**:

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

Mean, Variance, and Covariance, Correlation Coefficient

- Given feature-target data (x_i, y_i), $i = 1, 2, ..., N$
- Mean:

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i, \quad \bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i$$

- Variance:

$$\sigma_x^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2, \quad \sigma_y^2 = \frac{1}{N}\sum_{i=1}^{N}(y_i - \bar{y})^2$$

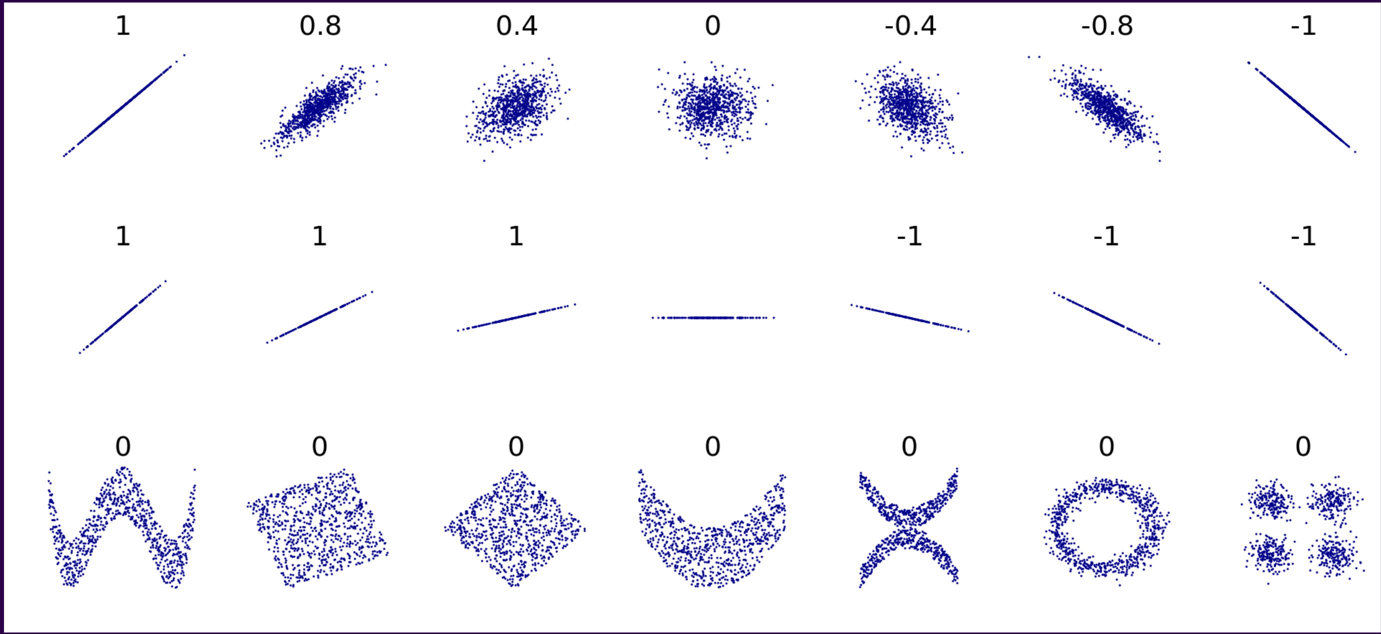
- **Covariance**:

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

- **Correlation Coefficient**:

$$\rho = \frac{\sigma_{xy}}{\sigma_x \sigma_y}$$

Lab: Gaining Intuition



https://en.wikipedia.org/wiki/Pearson_correlation_coefficient

Outline

1 Leftovers from Day 1

2 Introduction to Machine Leaning

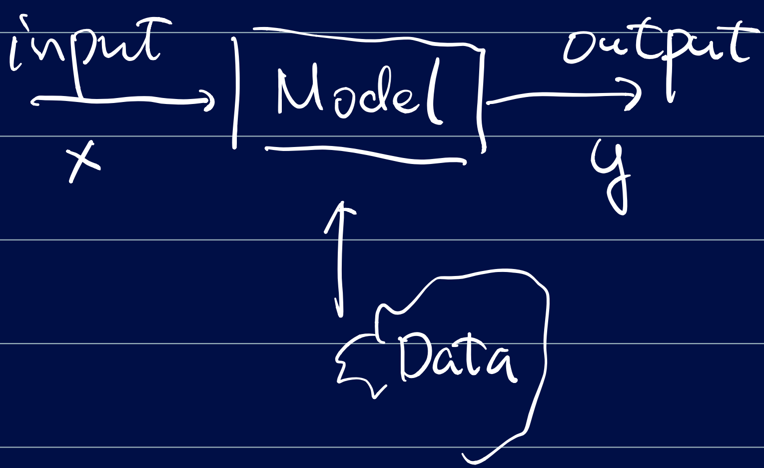
3 Lab: Simple Linear Model

4 Lab: Goodness of Fit

5 Statistics Basics

6 Least Squares Solution

Machine Learning



Ex: x : movie genre

y : IMDB score

Dataset: $\underbrace{(x_i, y_i)}_{\text{a sample}}$ $\quad i = 1, 2, \dots N$

We also call x features, y labels

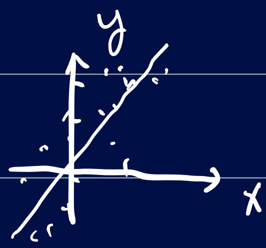
Task: to find a model $f(x)$ to predict

$$\hat{y} = f(x)$$

($\hat{y}$: "y hat", denotes the prediction given by the model)

Ex: $\hat{y} = f(x) = \underset{\sim}{w_1} x + \underset{\sim}{w_0}$

"linear model"



$w_0 = 0$

$w_1 = 2$

w_0, w_1 : parameters of the model

Cost functions : measure how well the model fits the data

MSE : $\frac{1}{N} \sum_{i=1}^{N} \| y_i - \hat{y}_i \|^2$

y_i: the labels in the dataset
"groud truth" label

$\hat{y}_i = f(x_i)$ predictions

Compute MSE in NumPy

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \quad \hat{y} = \begin{bmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_N \end{bmatrix} \quad \text{MSE}: \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

$$y - \hat{y} = \begin{bmatrix} y_1 - \hat{y}_1 \\ y_2 - \hat{y}_2 \\ \vdots \\ y_N - \hat{y}_N \end{bmatrix}$$

In NumPy, $z = (y - \hat{y})$ ** 2, this will square every element

$$\begin{bmatrix} (y_1 - \hat{y}_1)^2 \\ \vdots \\ (y_N - \hat{y}_N)^2 \end{bmatrix}$$

In our exercise $N = 506$

np.mean(z) will give $\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$

(For a vector z with N elements

np.mean(z) gives $\frac{1}{N}\sum_{i=1}^{N} z_i$)

LS Fit Solution

- Model:

$$\hat{y} = b + wx$$

LS Fit Solution

- Model:

$$\hat{y} = b + wx$$

- Optimization:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

LS Fit Solution

- Model:

$$\hat{y} = b + wx$$

- Optimization:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

- Solution:

$$\hat{y} = \bar{y} + \rho\frac{\sigma_y}{\sigma_x}(x - \bar{x})$$

LS Fit Solution

- Model:

$$\hat{y} = b + wx$$

- Optimization:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

- Solution:

$$\hat{y} = \bar{y} + \rho\frac{\sigma_y}{\sigma_x}(x - \bar{x})$$

$$w = \rho\frac{\sigma_y}{\sigma_x}, \quad b = \bar{y} - w\bar{x}$$

LS Fit Solution

- Model:

$$\hat{y} = b + wx$$

- Optimization:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

- Solution:

$$\hat{y} = \bar{y} + \rho\frac{\sigma_y}{\sigma_x}(x - \bar{x})$$

$$w = \rho\frac{\sigma_y}{\sigma_x}, \quad b = \bar{y} - w\bar{x}$$

- Prediction:

$$y_{new} = b + wx_{new}$$

LS Fit Solution

- Model:

$$\hat{y} = b + wx$$

- Optimization:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

- Solution:

$$\hat{y} = \bar{y} + \rho\frac{\sigma_y}{\sigma_x}(x - \bar{x})$$

$$w = \rho\frac{\sigma_y}{\sigma_x}, \quad b = \bar{y} - w\bar{x}$$

- Prediction:

$$y_{new} = b + wx_{new}$$

- Compute the LS fit model

NYU TANDON SCHOOL OF ENGINEERING

Linear Regression

- Today: linear models! In 1D, this is $f(x) = w_1 x + w_0$
- 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,

$$
\begin{aligned}
y_1 &\approx w_0 + w_1 x_1 \\
y_2 &\approx w_0 + w_1 x_2 \\
&\vdots \\
y_N &\approx w_0 + w_1 x_N.
\end{aligned}
$$

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),

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

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

NYU TANDON SCHOOL OF ENGINEERING

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 $\mathbf{w} = [w_0, w_1, w_2]^T$

  $y_1 \approx w_0 + w_1 x_{11} + w_2 x_{12}, \cdots$
- 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`