Supervised Learning

training data set $\Rightarrow$ $\mathcal{X} = \{(x_i, y_i)\}_{i=1}^{N}$

x_i → ith data point

y_i → ith label, ith output

Task: predicting whether $\underbrace{\text{a car}}_{x_i}$ is $\underbrace{\text{a family car or not}}_{y_i}$

$$x_i = \underbrace{\begin{bmatrix} x_{i1} \\ x_{i2} \end{bmatrix}}_{2\times 1} \begin{array}{l} \rightarrow \text{price} \\ \rightarrow \text{engine power} \end{array}$$

$$y_i = \begin{cases} 1 & \text{if } x_i \text{ is a family car} \\ 0 & \text{otherwise} \end{cases}$$

$$X = \begin{array}{c} \begin{array}{cc} \text{price} & \text{engine power} \end{array} \\ \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \\ x_{31} & x_{32} \\ \vdots & \vdots \\ x_{N1} & x_{N2} \end{bmatrix}_{N \times D} \end{array} \qquad y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_N \end{bmatrix}_{N \times 1}$$

$x_{31} \; x_{32}$ → car #3

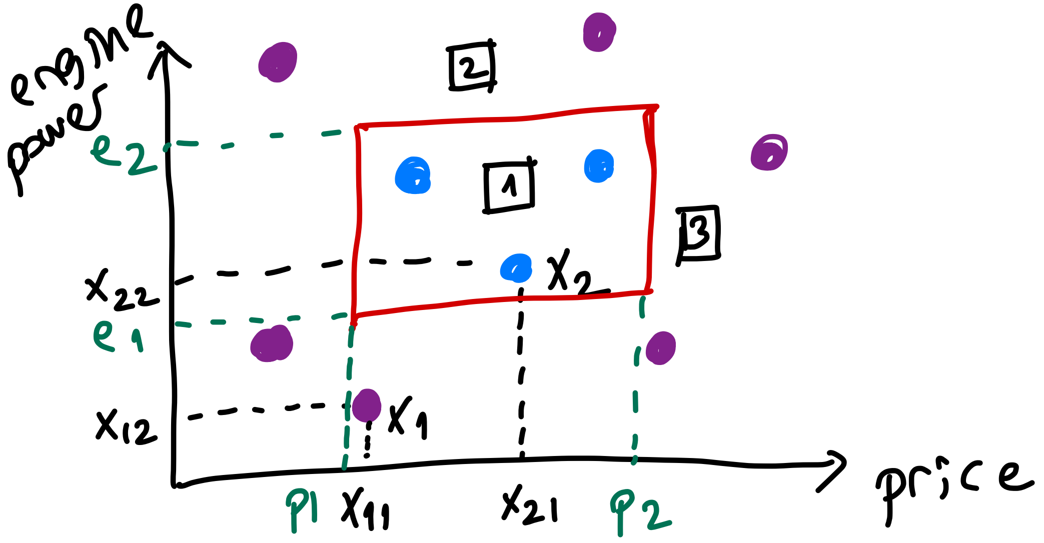
X → data matrix

y → label vector

N → # of data points

D → # of features

in our case $D = 2$



● : family car (positive)

● : other type of car (negative)

$$x_1 = \begin{bmatrix} x_{11} \\ x_{12} \end{bmatrix} \qquad y_1 = 0$$

$$x_2 = \begin{bmatrix} x_{21} \\ x_{22} \end{bmatrix} \qquad y_2 = 1$$

$$\theta = \{ p_1, \underline{\underline{p_2}}, e_1, \underline{\underline{e_2}} \}$$

RECTANGLES — model family

model parameters

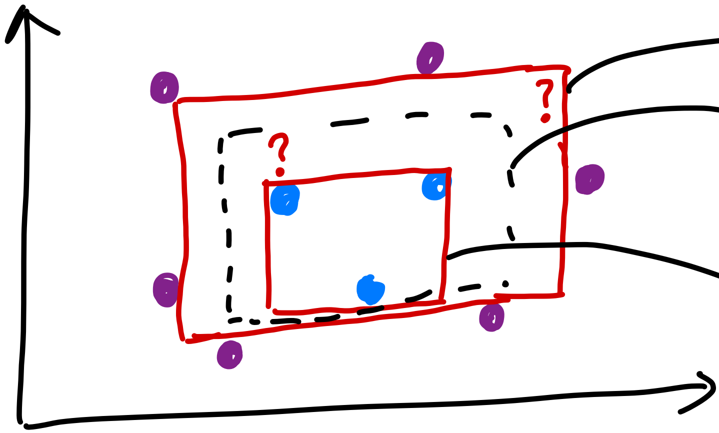
$$f(x_{N+1} \mid p_1, p_2, e_1, e_2) = ? \quad \Leftarrow \text{prediction}$$

LEARNING — Finding the best θ

$\boxed{1}$ $\underbrace{p_1 \leq x_{N+1,1} \leq p_2}_{\text{TRUE}}$ & $\underbrace{e_1 \leq x_{N+1,2} \leq e_2}_{\text{TRUE}}$ $\Rightarrow \hat{y}_{N+1} = 1$

$\boxed{2}$ $\underbrace{p_1 \leq x_{N+2,1} \leq p_2}_{\text{TRUE}}$ & $\underbrace{e_1 \leq x_{N+2,2} \leq e_2}_{\text{FALSE}}$ $\Rightarrow \hat{y}_{N+2} = 0$

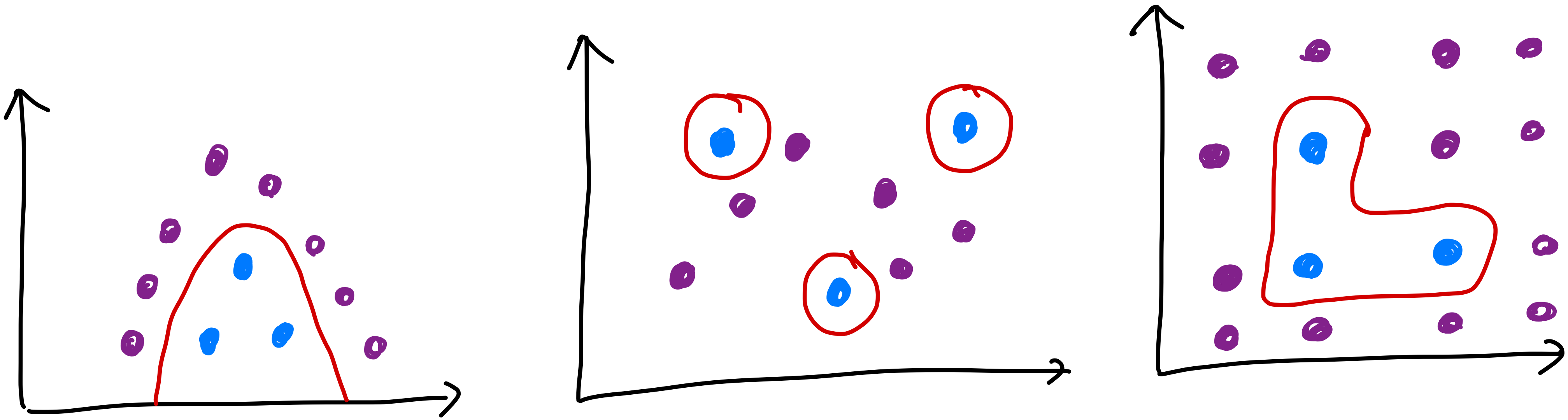
$\boxed{3}$ $\underbrace{p_1 \leq x_{N+3,1} \leq p_2}_{\text{FALSE}}$ & $\underbrace{e_1 \leq x_{N+3,2} \leq e_2}_{\text{TRUE}}$ $\Rightarrow \hat{y}_{N+3} = 0$



→ general solution

→ all rectangles in between are called "version space".

→ specific solution



Model Complexity ↑

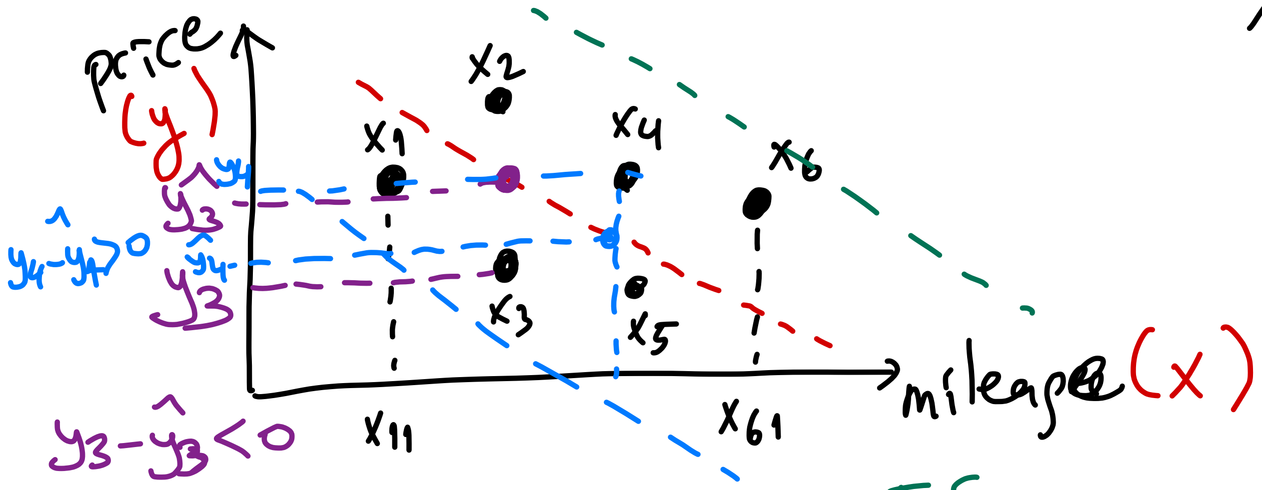
Prediction Performance

Training ↑

Test ↓

overfitting { At first both performances will improve, after some point test accuracy will get worse.

Linear Regression:



$$X = \begin{bmatrix} x_{11} \\ x_{21} \\ x_{31} \\ x_{41} \\ x_{51} \\ x_{61} \end{bmatrix} \quad 6 \times 1 \qquad y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{bmatrix} \quad 6 \times 1$$

N (rows), D (columns)

↳ output vector

SET OF LINES

Model Family

$$\theta = \{ w_1, w_0 \}$$

w_0 → bias

w_1 → slope

$$\hat{y}_i = w_1 . x_i + w_0$$

predicted outputs

$$\begin{cases} \hat{y}_1 = w_1 . x_1 + w_0 \\ \hat{y}_2 = w_1 . x_2 + w_0 \\ \vdots \\ \hat{y}_6 = w_1 . x_6 + w_0 \end{cases}$$

$$e_1 = y_1 - \hat{y}_1$$

$$e_2 = y_2 - \hat{y}_2$$

$$\vdots$$

$$e_6 = y_6 - \hat{y}_6$$

errors ↑ observed ↑ predicted ↑

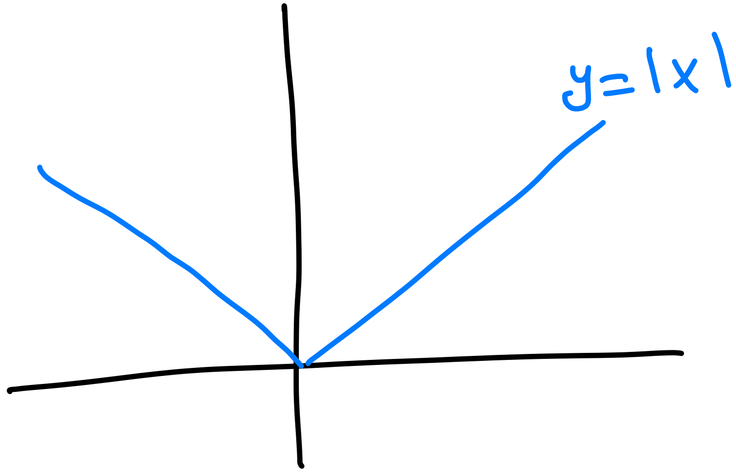
OPT 1 ✗

minimize $\sum_{i=1}^{N} (y_i - \hat{y}_i) = \sum_{i=1}^{N} e_i$

0

OPT 2 ✗

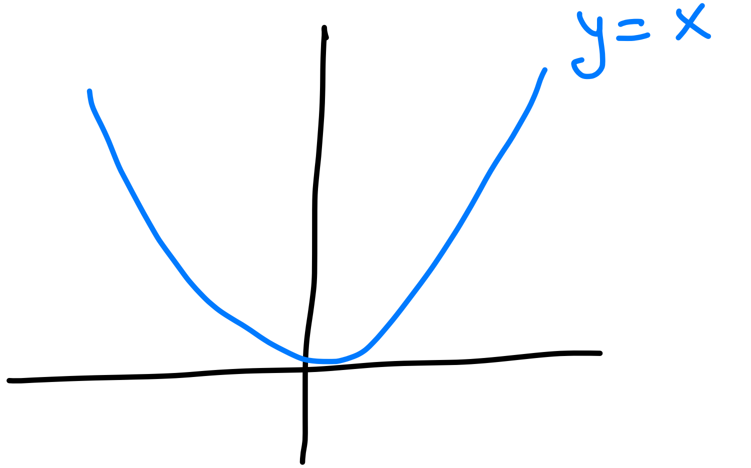
minimize $\sum_{i=1}^{N} |y_i - \hat{y}_i| = \sum_{i=1}^{N} |e_i|$



0

OPT 3 ✓

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



$e_i = y_i - \hat{y}_i$

$\hat{y}_i = w_1 x_i + w_0$

$\Rightarrow e_i = y_i - (w_1 x_i + w_0)$

$$\text{minimize} \sum_{i=1}^{N} \left[y_i - [\overset{?}{w_1} x_i + \overset{?}{w_0}] \right]^2$$

with respect to : w_1 and w_0

$$\text{Error}(w_0, w_1 | \mathcal{X}) = \sum_{i=1}^{N} (y_i - (w_1 x_i + w_0))^2$$

$$\frac{\partial \text{Error}}{\partial w_0} = \frac{\partial \left[\sum_{i=1}^{N} (y_i - (w_1 x_i + w_0))^2 \right]}{\partial w_0} = \sum_{i=1}^{N} \frac{\partial [y_i - (w_1 x_i + w_0)]^2}{\partial w_0}$$

$$= \sum_{i=1}^{N} \cancel{2} [y_i - (w_1 x_i + w_0)]^{\cancel{1}} \cdot \cancel{(-1)} \Rightarrow \boxed{\sum_{i=1}^{N} (y_i - w_1 x_i - w_0) = 0}$$

$$\frac{\partial \text{Error}}{\partial w_1} = \boxed{\sum_{i=1}^{N} (y_i - w_1 x_i - w_0) . x_i = 0}$$

Exercise: Solve for w_0 & w_1.

$$w_1 = \frac{\sum_{i=1}^{N} x_i . y_i - \left(\sum_{i=1}^{N} x_i/N\right)\left(\sum_{i=1}^{N} y_i/N\right) . N}{\sum_{i=1}^{N} x_i^2 - N . \left(\sum_{i=1}^{N} x_i/N\right)^2}$$

$$w_0 = \left(\sum_{i=1}^{N} y_i/N\right) - w_1 . \left(\sum_{i=1}^{N} x_i/N\right)$$

ML Algorithm

1. Collect data $\mathcal{X} = \{(x_i, y_i)\}_{i=1}^{N}$
2. pick a model family ← set of lines
3. pick a loss/error function ← sum of squared errors
4. learn the parameters on the training set ← calculus knowledge