Basic introduction to class + machine learning def and few examples

Def [machine learning](infromal):
A study of algorithms to improve the performance P at task T with experience E.

Example:

a) T: spam classification

b) P: accuracy

c) E: a set of mails labeled spams or not

Prediction: given an observation to a system, you get an output.
Examples: predict weather or a number in a pictue etc ...

Note:
Output $y \in \mathbb{R} \Longrightarrow$ repression problem.
Output y is in set/category (i.e one posible value out of set e.g {yes, no} or {1,2,..9}) $\Longrightarrow$ Classification problem.

Def [supervised learning]:
Given a set $\{x^{(i)})y^{(i)}\}_{i=1}^{N}$ of N data points, learn a prediction function:

$$f : x \to y$$

such that given a new x, f can accurately predict the correspendence y.

Examples:
find a cut curve. our function is $y = sin(2\pi x)$
Let's consider a simple model:

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

This model only fit linear fucntions (lines). and we need w_0.
$w = (w_0, w_1)$. $\hat{y}(x; w) = w_0 + w_1 x$. This is an example of a linear model (in terms of parameter not function.)

Linear model: The model is linear to the parameters. they have nice properties.

How can we learn the parameters w_0 and w_1?
$\to$ this can be done by minimizing the error margin using following formula:

$$E(w) = \frac{1}{2}\Sigma_{n=1}^{N}\{\hat{y}(x^{(n)}; w) - y^(n))\}^2$$

now, using it in our case:

$$E(w) = \frac{1}{2}\Sigma_{n=r}^{N}\{w_0 + w, y^{(n)} - x^{(n)}\}^2$$

The error function is quadratic in terms of parameters. giving a linear derivative.
$\rightarrow$ min E(w) has a min value that helps us find w_0 and w_1.

Our model is linear however we have a sign function, hence our "approximation" is not good.

We can assume higher order (N)

$$\hat{y}(x; y) = \Sigma_{i=0}^{N} w_i x^i$$

When we try 3 degree polynomial, we get good a model that fits the curve.(good solution + small error)
but with degree 9, we get 0 errors but a solution that badly fits the curve.