Let $g = \mathcal{A}(\mathcal{D}, \mathcal{H})$. Predict $\hat{y}^* = g(\vec{x}^*)$. What if g found the closest $\vec{x}_i \in \mathcal{D}$ to $\vec{x}^*$ and returned $\hat{y}^* = y_i$? This closest $\vec{x}_i$ is called its neighbor. By closest, we need a notion of a difference between two observations:

$$d(\vec{x}_i \vec{x}_k) = \|\vec{x}_i - \vec{x}_k\|^2 = (\vec{x}_i - \vec{x}_k)^T(\vec{x}_i - \vec{x}_k) = \sum_{j=1}^{p}(x_{ij} - x_{kj})^2$$

This is called $L1$ distance or Euclidean norm squared distance. Here, $\mathcal{H}$ and $\mathcal{A}$ are difficult to define.
What if g located the closest $\vec{x}_i$?. Then $\hat{y} = \text{Mode}[y_1, \ldots, y_k]$ where each y_i represents the nearest x_i's. This is called the K nearest neighbors, or KNN algorithm . The weakness of this algorithm is when p is large, where there are too many dimensions and so not all x_j terms are equally predictive. In this algorithm, k and d must be chosen.

So far, we have only been concerned with $y = \left\{0, 1\right\}$. This is called "binary classification." Tif $y = \left\{0, 1, \ldots, k\right\}$, where the response level are nominal (no order), there is called "classification" or "multi-level classification."
What if $y \in \mathbb{R}$ or $y \in R \subseteq \mathbb{R}$? This is called "regression." The threshold, perception and SVM cannot do regression without some adaptations.

Null Model: doesn't care about $\vec{x}_i$s. Therefore $g(\vec{x}) = \bar{y} = \frac{1}{n}\sum_i y_i$.
Linear Regression Model: Consider $\mathcal{H} = \left\{\vec{w} \cdot \vec{x} : \vec{w} \in \mathbb{R}^{p+1}\right\}$ where $\vec{x} = \begin{bmatrix} 1 & x_1 & \ldots & x_p \end{bmatrix}$ and $\vec{w} = \begin{bmatrix} w_0 & w_1 & \ldots & w_p \end{bmatrix}$. Then we get

$$H = \left\{w_0 + w_1 x_1 + \cdots + w_p x_p : w_0 \in \mathbb{R}, w_1 \in \mathbb{R}, \ldots, w_p \in \mathbb{R}\right\}$$

Here $\vec{w}$ is the linear coefficients. The dimension of this parameter space is $p + 1$.
Imagine this for for $p = 1$ care.

$$\mathcal{H} = \left\{w_0 + w_1 x_1 : w_0 \in \mathbb{R}, w_1 \in \mathbb{R}\right\}$$

Then the candidate in $\mathcal{H}$ that most closely resembles f is

$$h^*(\vec{x}) = w_0^* + w_1^* = \beta_0 + \beta_1 x$$

What about the errors?

$$y = h^*(\vec{x}) + \varepsilon = h^*(\vec{x}) + \overbrace{(t(\vec{z}) - f(\vec{x}))}^{\text{ignorance}} + \underbrace{(f(\vec{x}) - h^*(\vec{x}))}_{\text{misspecification of linear model}}$$

where ε is the noise or error. Note that h^* is inaccessible since we have to make an imperfect fit with finite data. Therefore

$$y = g(\vec{x}) + e = g(\vec{x}) + \underbrace{(t(\vec{z}) - f(\vec{x})) + (f(\vec{x}) - h^*(\vec{x}))}_{\varepsilon} + \overbrace{(h^*(\vec{x}) - g(\vec{x}))}^{e-\varepsilon}$$

where $e - \varepsilon$ is the estimation error. We call e the residuals.
As $n \to \infty$, $g(\vec{x}) \to h^*(\vec{x})$ and $e - \varepsilon \to 0$. But $y \neq g(\vec{x})$ since the other two errors are still present.

For the linear model for $p = 1$, we need a loss function to fit $\vec{w}$. Recall the sum of squared error formula. Let's do some manipulations to it.

$$\begin{aligned}
SSE &= \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \sum e_i^2 \\
&= \sum_{i=1}^{n}(y_i - (w_0 + w_1 x_i))^2 \\
&= \sum_{i=1}^{n} y_i^2 + w_0^2 + w_1^2 x_i^2 - 2y_i w_0 - 2y_i w_1 x_i + 2w_0 w_1 x_i \\
&= \sum y_i^2 + n w_0^2 + w_1^2 \sum x_i^2 - 2n\bar{y}w_0 - 2w_1 \sum x_i y_i + 2w_0 w_1 n\bar{x}
\end{aligned}$$

Choose w_0 and w_1 to minimize the above.

$$\frac{\partial}{\partial w_0} SSE = 2nw_0 - 2n\bar{y} + 2w_1 n\bar{x} = 0 \to \hat{w}_0 = \bar{y} - w_1\bar{x}$$

$$\begin{aligned}
\frac{\partial}{\partial w_1} SSE &= 2w_2 \sum x_i^2 - 2\sum y_i x_i + 2w_0 n\bar{x} = 0 \\
&= w_1 \sum x_i^2 - \sum y_i x_i + (\bar{y} - w_1\bar{x})n\bar{x} = 0 \\
&= w_1 \sum x_i^2 - \sum y_i x_i + n\bar{x}\bar{y} - w_1 n\bar{x}^2 = 0 \\
w_1(\sum x_i^2 - n\bar{x}^2) &= \sum y_i x_i - n\bar{x}\bar{y} \\
\hat{w}_1 &= \frac{\sum y_i x_i - n\bar{x}\bar{y}}{\sum x_i^2 - n\bar{x}^2} = \frac{(n-1)S_{xy}}{(n-1)S_x^2} = r\frac{S_y}{S_x} \\
\hat{w}_0 &= \bar{y} - r\frac{S_y}{S_x}\bar{x} = \beta_0
\end{aligned}$$