For supervised learning, you need $\mathcal{D}, \mathcal{H}, \mathcal{A}$ where $g = \mathcal{A}(\mathcal{D}, \mathcal{H})$.
How to get prediction? Let $\mathcal{D} = \left\{\langle \vec{x}_i, y_i \rangle\right\}$ where $i - 1, \ldots, n$. Your estimate is $\hat{y}_i = g(\vec{x}_i)$. It is also called an in-sample fit.
We want $\left\{\hat{y}_1, \ldots, \hat{y}_n\right\}$, a set of in-sample fits / predictions to be $\approx \left\{y_1, \ldots, y_n\right\}$, which is good but impossible due to 3 errors.
How to predict for new data/observation $\vec{x}^*$? Let $\hat{y}^* = g(\vec{x}^*)$.

Let $y \in \left\{0, 1\right\}$. Let's use only $x =$ salary. Then

$$\mathcal{H} = \left\{\mathbb{1}_{x > x_T} : x_T \in \mathbb{R}\right\}$$

, a family of step functions with a break at x_T, the parameter or threshold.
A more flexible model family to use is where there are multiple floor and ceiling functions put together on $\mathbb{R}$; however this is more complicated. Also, we still need $\mathcal{A}$, an algorithm.

Let's define some error functions $\text{Err}(\vec{y}, \vec{\hat{y}}) > 0 \forall$ inputs.

- Mean of absolute error: $MAE = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i|$
  This is also called misclassification error and can be written as $\frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{y_i \neq \hat{y}_i}$
- Sum of absolute error: $SAE = \sum_{i=1}^{n} |y_i - \hat{y}_i|$
- Sum of squared error: $SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$
- Mean of squared error: $MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$

Let

$$SSE = \sum_{i=1}^{n} (y_i - h(\vec{x}_i) = \overset{\text{argmin}}{h \in \mathcal{H}} \left\{SSE(h)\right\}$$

This is the same as saying

$$x_T^* = \overset{\text{argmin}}{x_T} \left\{\sum_{i=1}^{n} (y_i - \mathbb{1}_{\vec{x}_T - x_T})^2\right\}$$

In this case, $\mathcal{A}$ would be a greedy search - tries every possibility.

Let X_1, X_2 be continuous. Choose a model such that

$$\mathcal{H} = \left\{\mathbb{1}_{x_1 > x_{T_1}} \mathbb{1}_{x_2 > x_{T_2}} : \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in \mathbb{R}^2\right\}$$

This would be a bad model because it would only capture values inside and outside a right angle.

A better model would be one that captures values above or below a certain line.

$$\begin{aligned}
H &= \left\{ \mathbb{1}_{x_2 > a + bx_i : \begin{bmatrix} a \\ b \end{bmatrix} \in \mathbb{R}^2} \right\} \\
&= \left\{ \mathbb{1}_{a + bx_1 - x_2 > 0} : \begin{bmatrix} a \\ b \end{bmatrix} \in \mathbb{R}^2 \right\} \\
&= \left\{ \mathbb{1}_{-a - bx_1 + x_2 > 0} : \begin{bmatrix} a \\ b \end{bmatrix} \in \mathbb{R}^2 \right\} \\
&= \left\{ \mathbb{1}_{w_0 + w_1 x_1 + w_2 x_2 > 0} : \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix} \in \mathbb{R}^3 \right\} \\
&= \left\{ \mathbb{1}_{x_0 + \vec{w}\vec{x} > 0} : w_0 \in \mathbb{R}, \vec{w} \in \mathbb{R}^2 \right\}
\end{aligned}$$

This is called a linear classifier.
Let $\vec{w} = \begin{bmatrix} 1 & x_1 & x_2 \end{bmatrix}$ where we augment $\vec{x}$ to include a 1 in the first position, then we get

$$\mathcal{H} = \left\{ \mathbb{1}_{\vec{w} \cdot \vec{x} > 0} : \vec{w} \in \mathbb{R}^3 \right\}$$

Use the same error function SSE

$$g =^{\text{argmin}} \left\{ SSE(h) \right\} \iff \vec{w}^* =^{\text{argmin}} \left\{ \sum_{i=1}^{n} (\mathbb{1}_{\vec{w} \cdot \vec{x} > 0})^2 \right\}$$