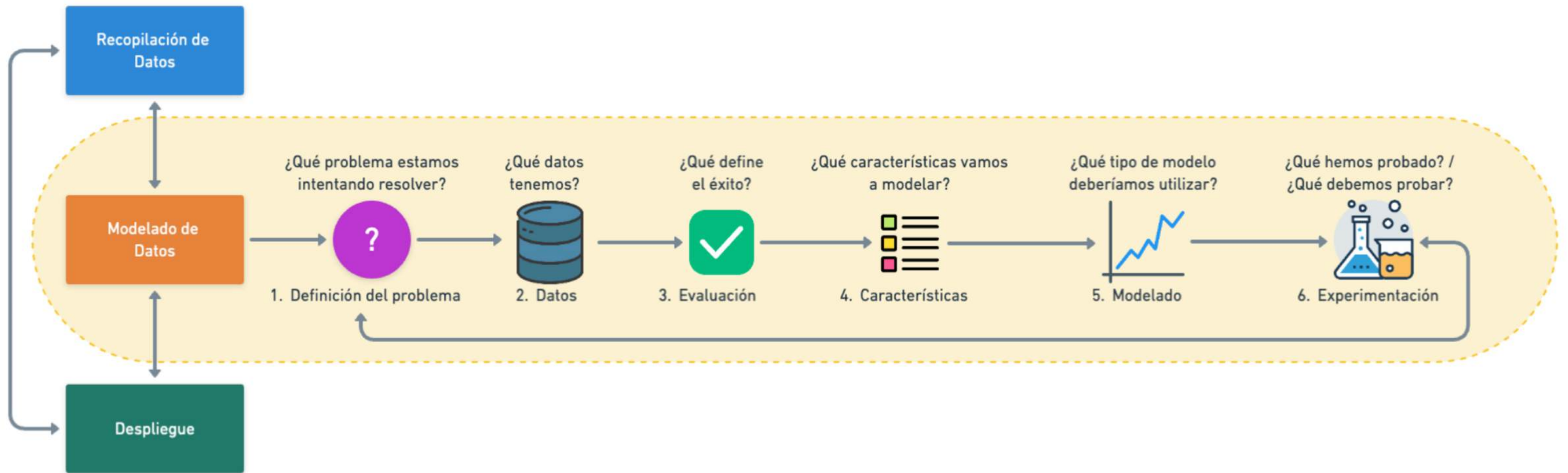


# MACHINE LEARNING

MTRO. ALFONSO GREGORIO RIVERO DUARTE

# FRAMEWORK



# MODELOS DE REGRESIÓN

Los modelos de regresión (tanto lineal como no lineal) se utilizan muchísimo para predecir valores numéricos como por ejemplo el sueldo. Si nuestra variable independiente es tiempo entonces podemos hacer predicciones de valores futuros, sin embargo nuestro modelo puede predecir también valores desconocidos del presente.

Las técnicas de Regresión son muy variadas, desde la regresión lineal hasta la SVR o la Regresión con Bosques Aleatorios.





# TÉCNICAS O ALGORITMOS

# MODELOS DE REGRESIÓN

- 1.Regresión Lineal Simple
- 2.Regresión Lineal Múltiple
- 3.Regresión Polinómica
- 4.Regresión con Máquinas de Soporte Vectorial (SVR)
- 5.Regresión con árboles de decisión
- 6.Regresión con bosques aleatorios



# REGRESIÓN LINEAL

# MODELOS DE REGRESIÓN

1. Idea de la Regresión Simple
2. Descripción del problema
3. Algoritmo de Regresión Lineal Simple
4. Evaluación

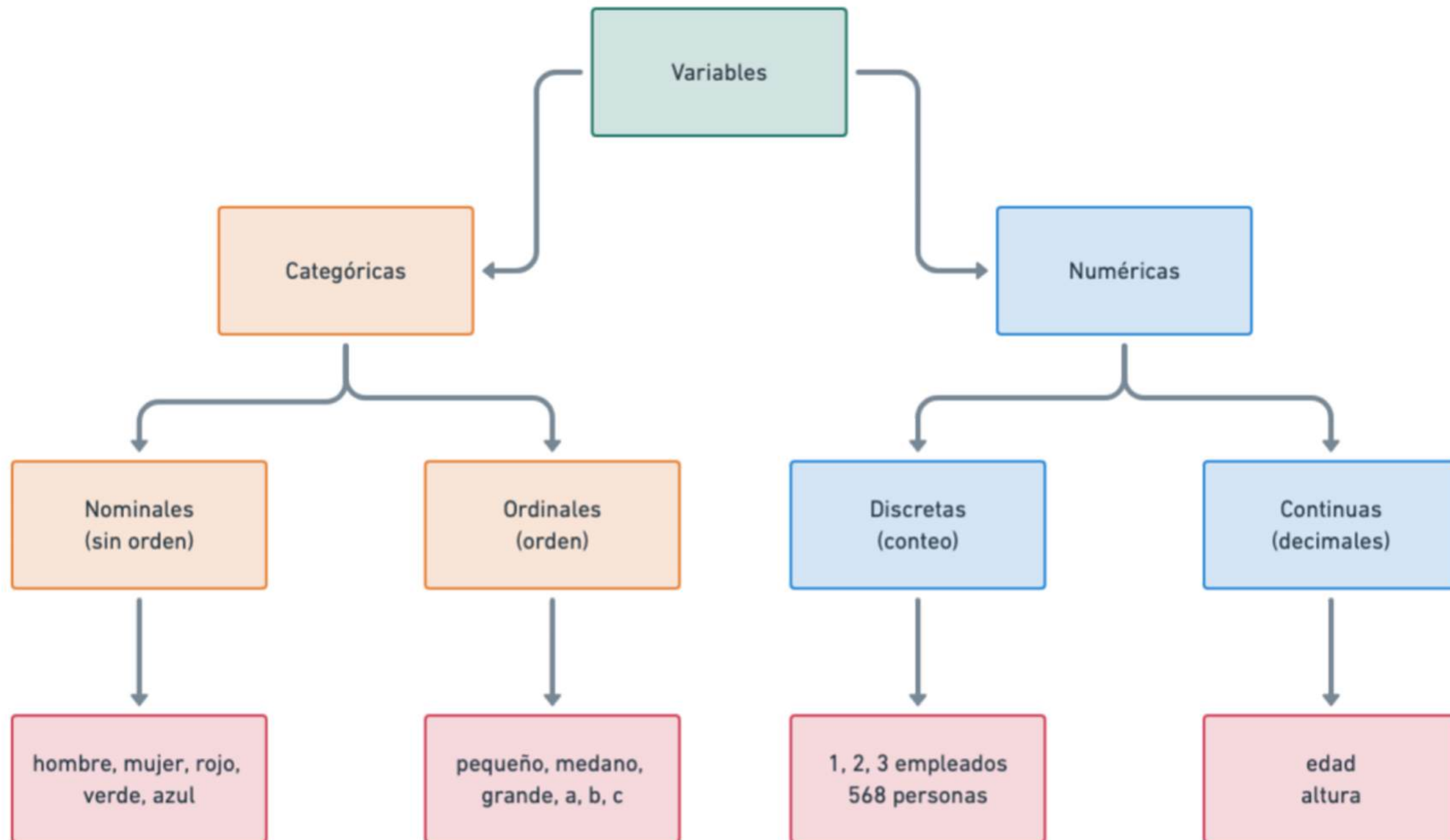




# IDEA DE LA REGRESIÓN SIMPLE



# VARIABLES



# REGRESIÓN LINEAL

En estadística, se llama análisis de la regresión al proceso estadístico de estimar las relaciones que existen entre variables.

...

Se centra en estudiar las relaciones entre una variable dependiente de una o más variables independientes.

# REGRESIÓN LINEAL



Crear una línea para  
predecir valores



Fácil de visualizar



Puedes dibujar una  
línea para ver cómo  
predice el resultado



# REGRESIÓN LINEAL

Regresión Lineal  
Simple

Diagram illustrating the components of the simple linear regression equation:

$$y = b_0 + b_1 * x_1$$

Labels and arrows:

- Constante (points to  $b_0$ )
- Coeficiente (points to  $b_1$ )
- Variable Dependiente (VD) (points to  $y$ )
- Variable Independiente (VI) (points to  $x_1$ )

Regresión Lineal  
Múltiple

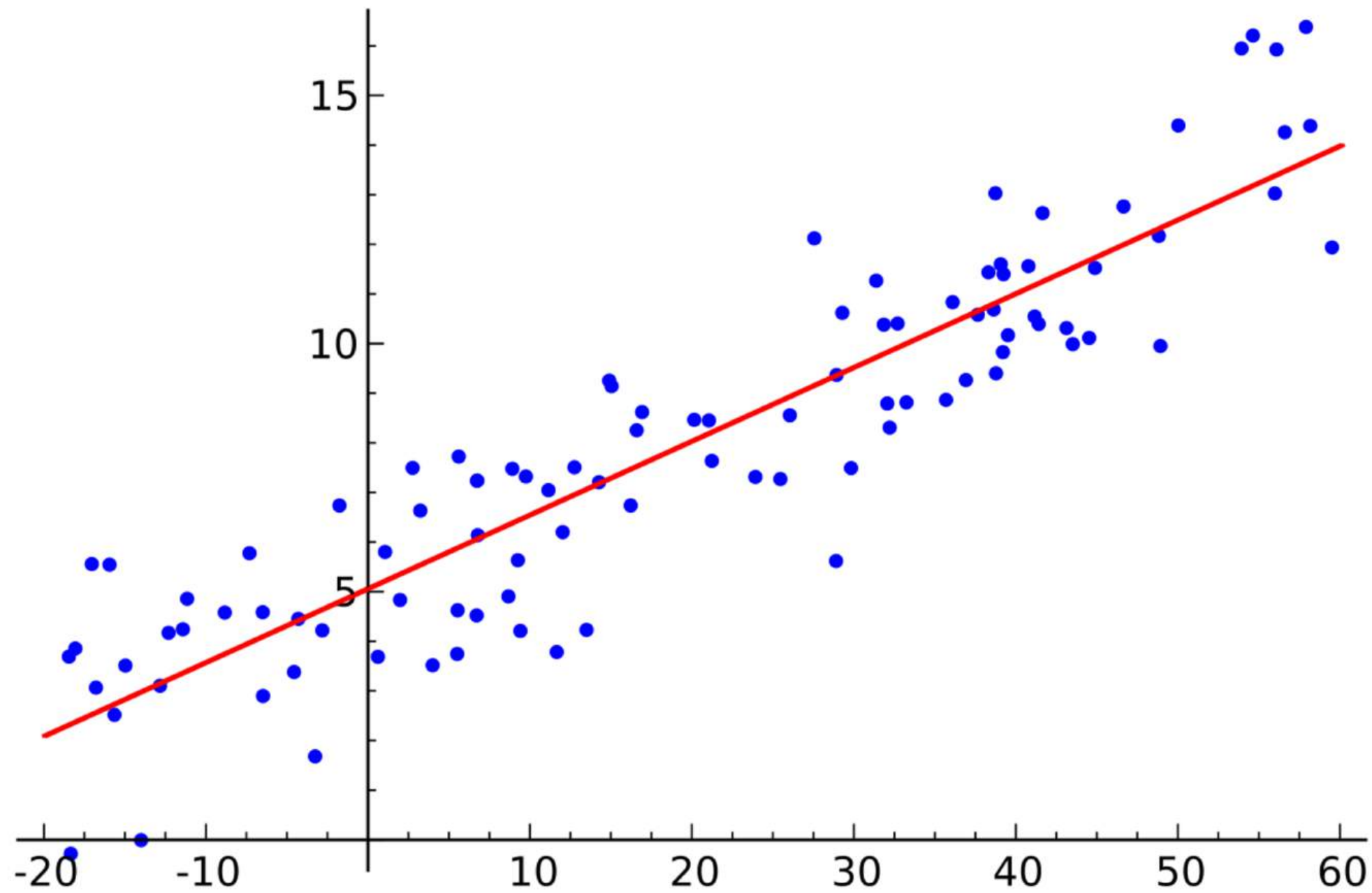
Diagram illustrating the components of the multiple linear regression equation:

$$y = b_0 + b_1 * x_1 + b_2 * x_2 + \dots + b_n * x_n$$

Labels and arrows:

- Variable Dependiente (VD) (points to  $y$ )
- Variables Independientes (VIs) (points to  $x_1, x_2, \dots, x_n$ )
- Constante (points to  $b_0$ )
- Coeficientes (points to  $b_1, b_2, \dots, b_n$ )

# REGRESIÓN LINEAL



$$Y \approx \beta_0 + \beta_1 X.$$

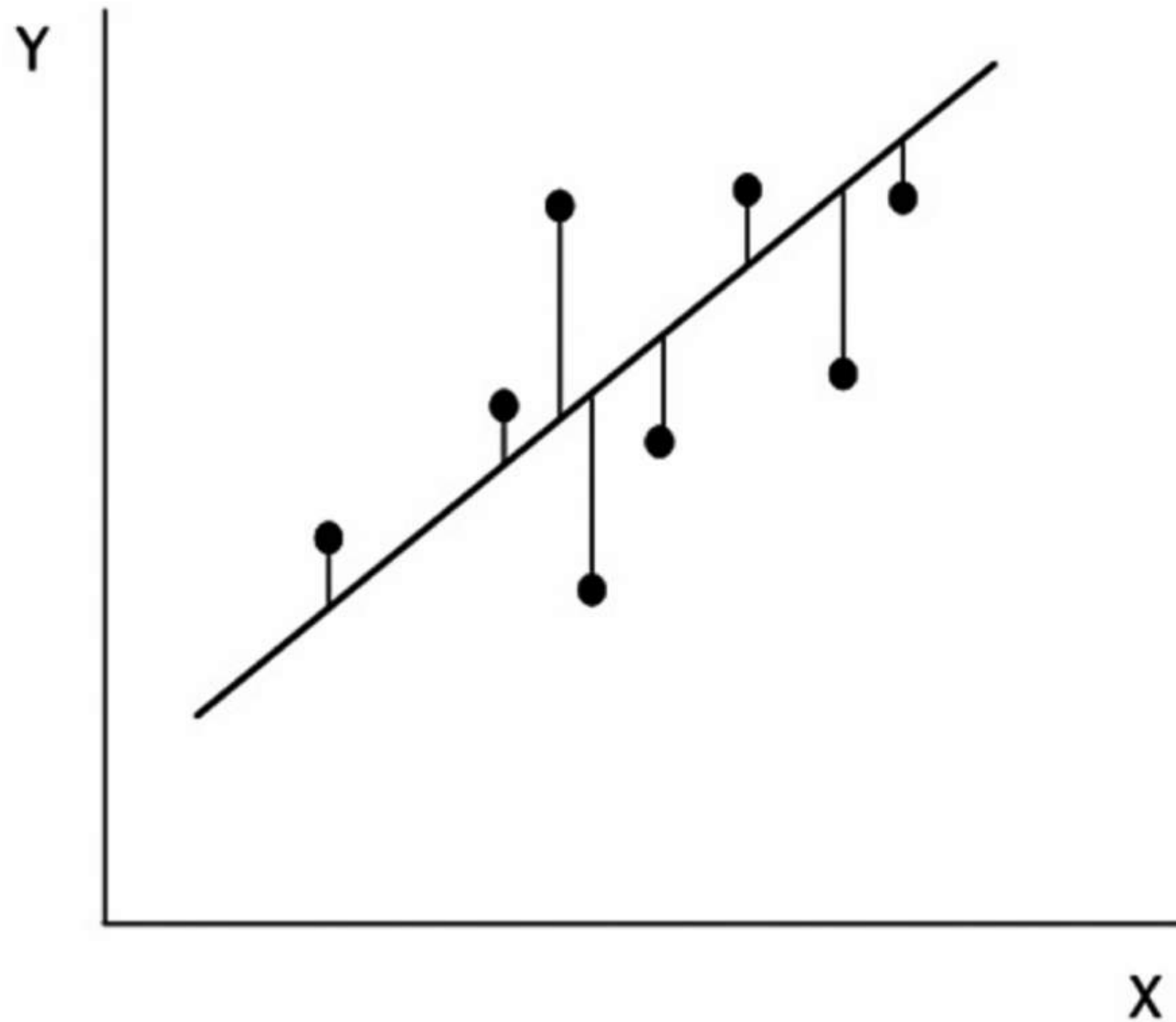
# REGRESIÓN LINEAL

La relación lineal entre **dos variables**  $x$  (variable independiente) y  $y$  (variable dependiente) se representa matemáticamente a través de la ecuación  $y=wx+b$ , donde  $w$  representa la pendiente (inclinación) de la línea recta y  $b$  es la intersección con el eje  $y$ .

Así, el objetivo de la Regresión Lineal es encontrar los valores más adecuados de  $w$  y  $b$  que representen de la mejor forma posible la relación entre las variables  $x$  y  $y$ .



# REGRESIÓN LINEAL



Aprendizaje **supervisado**

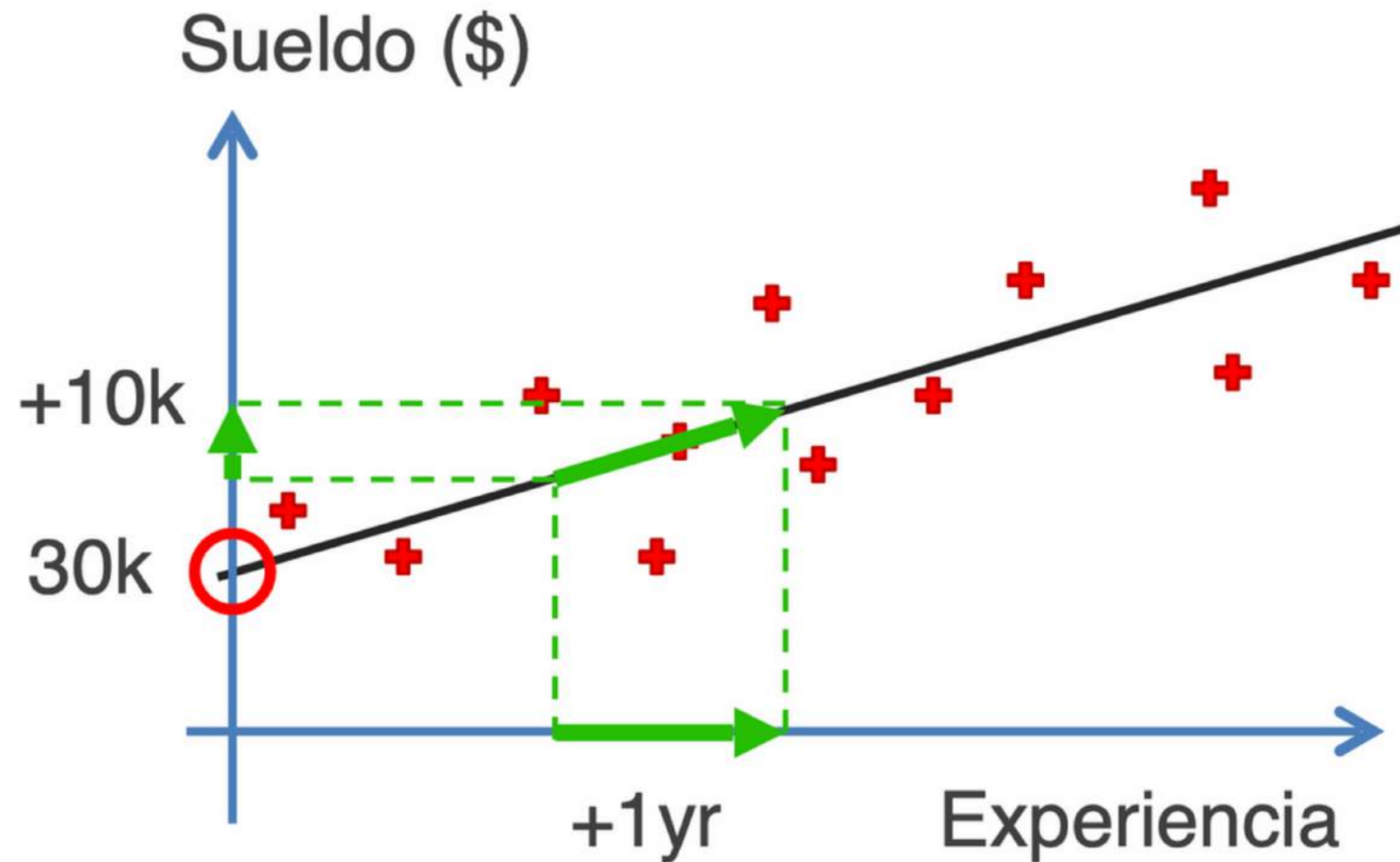
Aprendizaje **basado en modelos**

Se corresponde con un **modelo lineal**

Realiza predicciones computando una **suma ponderada de las características de entrada** y sumándole una constante conocida como **bias**

Intenta predecir **valores continuos**

# REGRESIÓN LINEAL



$$y = b_0 + b_1 * x$$

↓

$$\text{Sueldo} = b_0 + b_1 * \text{Experiencia}$$

Aca tenemos analizados la data de nuestro dataset, sueldo y experiencia.  
Nuestro modelo de regresión lineal es

$$\text{Sueldo} = b_0 + b_1 * \text{Experiencia}$$

# REGRESIÓN LINEAL

La regresión lineal simple nos devuelve precisamente la mejor recta, el mejor modelo lineal que se ajusta a los puntos.

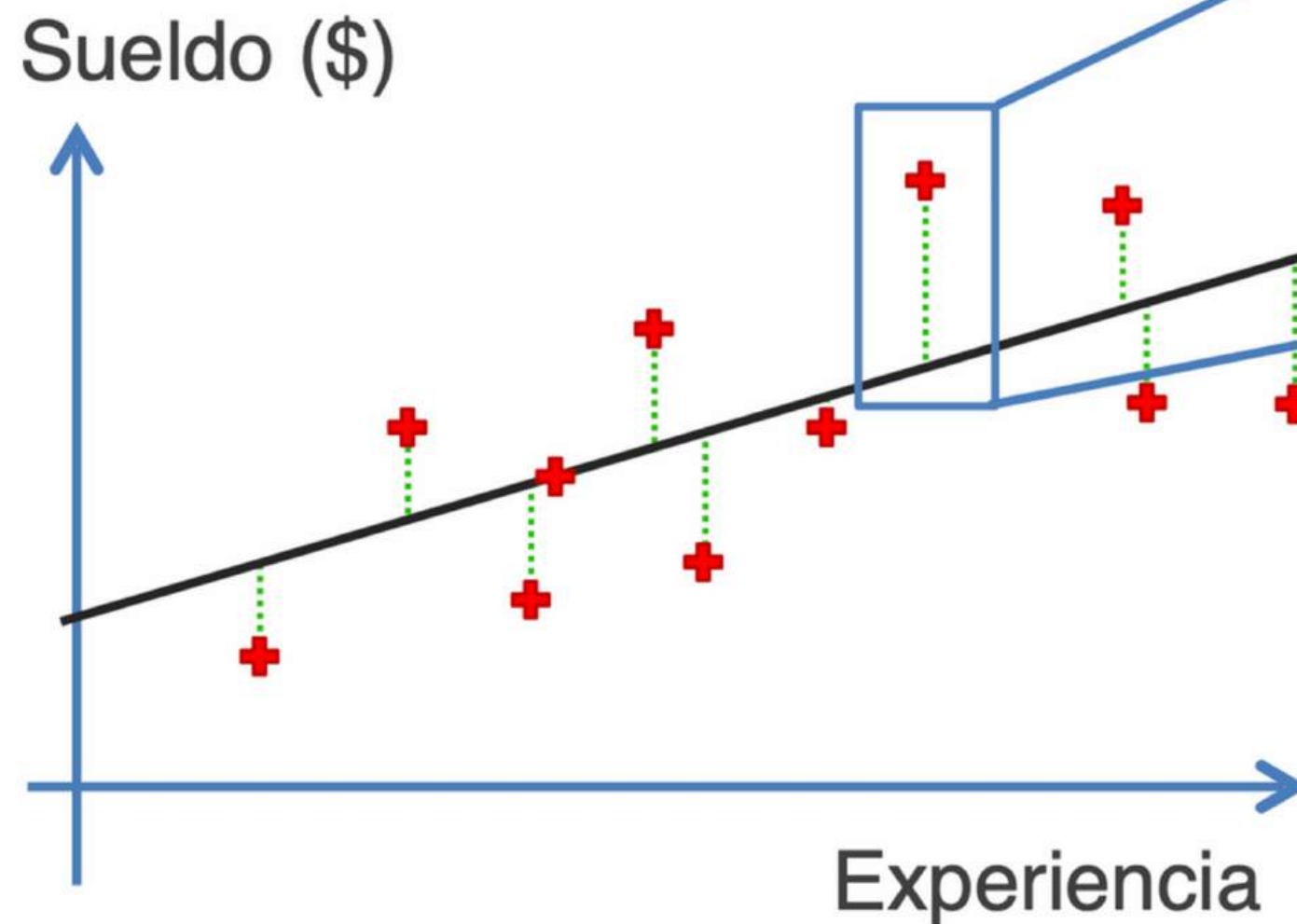
**$b_0$** : el valor en este caso del sueldo cuando la experiencia fuera cero años. (vértice origen)

**$b_1$** : tenemos el coeficiente de la pendiente. Nos indica cómo de inclinada es la recta cuando la experiencia es cero. Hay muchísimas hacia arriba y hacia abajo, esos son nuestros puntos. Algunas crecerán más rápido. Otras crecerán más despacio.



# MINIMOS CUADRADOS

## Regresión Lineal Simple



$$\min \sum_i (y_i - \hat{y}_i)^2$$

Queremos predecir el sueldo del empleado en función de sus años de experiencia.

# REGRESIÓN LINEAL

Dibujamos unas líneas verticales desde las observaciones reales hasta el modelo de regresión, hasta la recta. Esas líneas de puntos representan las diferencias entre lo que dice nuestro modelo y el dato real que tenemos.

La diferencia entre el dato real y lo que nos dice el modelo, esa discrepancia va a ser lo importante.

La cruz roja es el dato real. La verde es la predicción que predice el modelo en base a las observaciones que tenemos. Esa línea es lo que el empleado esta ganando de más o de menos.

De todas las rectas que se pueden calcular, siempre voy a elegir aquella que minimiza las distancias de los puntos rojos hacia la recta. El algoritmo suma todas esas diferencias. Y nos debemos quedar con la que minimice la suma de las lineas discontinuas de puntos.



# DESCRIPCIÓN DEL PROBLEMA



# REGRESIÓN LINEAL

Para este ejercicio estaremos utilizando un dataset muy interesante.

Se trata de dos columnas en el CSV, donde en la primera tenemos años de experiencia (podemos imaginar que se trata de los datos de una empresa, de una compañía, donde los empleados van acumulando años de experiencia) y en la columna de al lado tenemos el sueldo que cobra ese empleado (podemos suponer que son dólares).

Entonces para leerlo, cada fila se correspondería con un empleado o datos de empleados que, por ejemplo, el primero de ellos lleva 1.1 años de experiencia en la empresa y cobra 39.343. Así sucesivamente con todos y cada uno de ellos.

# REGRESIÓN LINEAL

La idea es que vamos a intentar predecir y crear un modelo lineal, lo que llamamos una **regresión lineal simple**, el cual se basa en **buscar de todas las rectas posibles, de todos los modelos lineales, todas las rectas, cuál es la que mejor se ajusta a estos datos?**

Cuál es la recta que sin llegar a pasar por todos los puntos porque probablemente no estén alineados, por lo menos es capaz de acercarse o de minimizar la distancia de la propia recta a los puntos.

# REGRESIÓN LINEAL

1

```
from sklearn.linear_model import LinearRegression
```

2

```
regressor = LinearRegression()
```

3

```
regressor.fit(X_train, y_train)
```

4

```
y_pred = regressor.predict(X_test)
```





# ALGORITMO



# RETO

## Reto:

Comencemos con:

- Active su ambiente local creado con miniconda en su equipo o su código de COLAB
- Abra el archivo [O6\\_Simple\\_LinearRegression.ipynb](#)
- Ejecute el código y analícelo junto al instructor

## Resultado Esperado:





# EVALUACIÓN

# PREGUNTAS Y RESPUESTAS

Mtro. Alfonso Gregorio Rivero Duarte

## Senior Data Manager – CBRE

**(+52) 5528997069**

**devil861109@gmail.com**

**<https://www.linkedin.com/in/alfonso-gregorio-rivero-duarte-139a9225/>**

