



**CLUB DE DESARROLLO
DE VIDEOJUEGOS
E INTELIGENCIA ARTIFICIAL**

PRE-PROCESAMIENTO



- Importar las librerías.
- Importar el dataset.
- Datos faltantes (Missing Data).
- Variables categóricas.
- Dividir el dataset en training set y test set.
- Escalamiento.

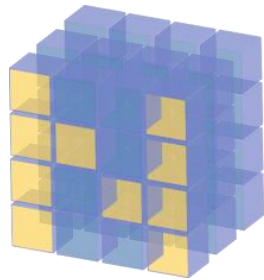
IMPORTAR LIBRERÍAS

```
import numpy as np
```

```
import pandas as pd
```

```
import matplotlib.pyplot as plt
```

matplotlib



NumPy

Pandas



IMPORTAR DATASET

- Ubicarnos en el directorio correcto.
- Importar el dataset.

```
dataset = pd.read_csv('Preprocesamiento.csv')
```

- Separar las variables dependientes de las independientes.

```
X = dataset.iloc[:, :-1].values
```

```
y = dataset.iloc[:, 3].values
```

DATOS FALTANTES (MISSING DATA)

- Importar la clase para manejo de datos faltantes.

```
from sklearn.preprocessing import Imputer
```

- Crear un objeto de la clase con los parámetros necesarios.

```
imputer = Imputer(missing_values='NaN', strategy = 'mean',axis = 0)
```

- Aplicar los métodos de la clase a nuestros datos.

```
imputer = imputer.fit(X[:,[1,2]])
```

```
X[:,[1,2]] = imputer.transform(X[:,[1,2]])
```

VARIABLES CATEGÓRICAS

```
from sklearn.preprocessing import LabelEncoder, OneHotEncoder  
labelencoder_X = LabelEncoder()  
X[:,0] = labelencoder_X.fit_transform(X[:,0])  
onehotencoder = OneHotEncoder(categorical_features=[0])  
X = onehotencoder.fit_transform(X).toarray()  
labelencoder_y = LabelEncoder()  
y = labelencoder_y.fit_transform(y)
```

DUMMY ENCODING

ID	Gender
1	Male
2	Female
3	Not Specified
4	Not Specified
5	Female



ID	Male	Female	Not Specified
1	1	0	0
2	0	1	0
3	0	0	1
4	0	0	1
5	0	1	0

TRAINING SET Y TEST SET

- Importar la librería.

```
from sklearn.model_selection import train_test_split
```

- Definir los tamaños del training y test set

```
X_train , X_test, y_train, y_test = train_test_split(X, y, test_size =  
0.25, train_size=0.75)
```


ESCALAMIENTO

Estandarización

$$z_i = \frac{x_i - \bar{x}}{\sigma}$$

Normalización

$$z = \frac{x - \min(x)}{\max(x) - \min(x)}$$

ESCALAMIENTO (ESTANDARIZACIÓN)

- Importar la clase para realizar escalamiento.

```
from sklearn.preprocessing import StandardScaler
```

- Crear un objeto de la clase con los parámetros necesarios.

```
scale_X = StandardScaler()
```

- Aplicar los métodos de la clase a nuestros datos.

```
X_train = scale_X.fit_transform(X_train)
```

```
X_test = scale_X.transform(X_test)
```

ESCALAMIENTO (NORMALIZACIÓN)

```
from sklearn.preprocessing import Normalizer
```

```
scale_X = Normalizer()
```

```
X_train = scale_X.fit_transform(X_train)
```

```
X_test = scale_X.transform(X_test)
```

REGRESIÓN LINEAL SIMPLE

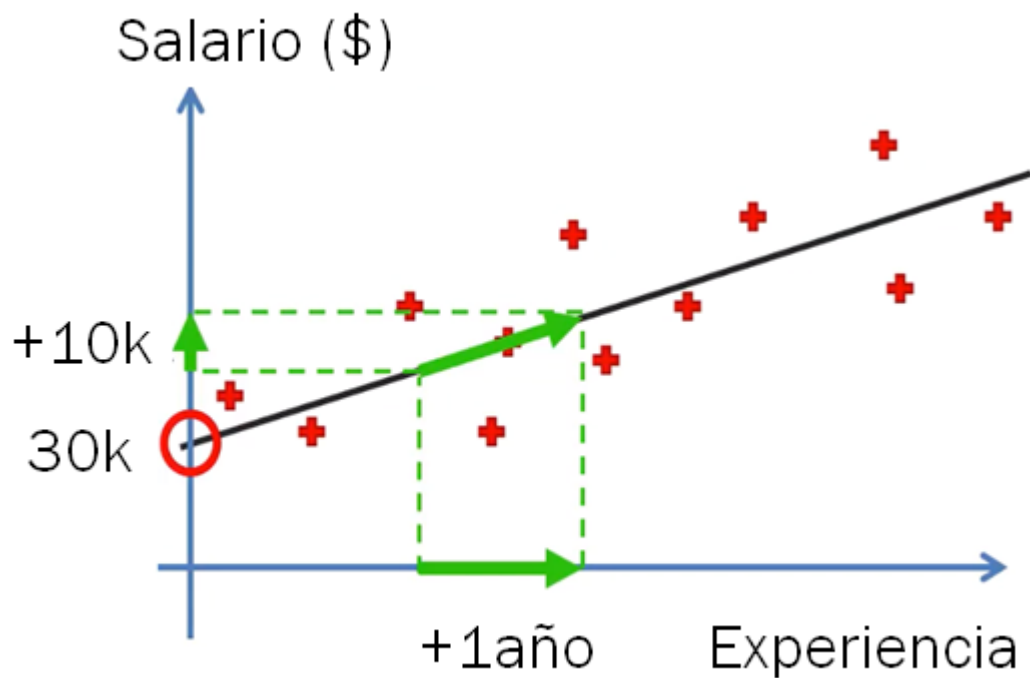
Constant

Coefficient

$$y = b_0 + b_1 * x_1$$

Dependent variable (DV)

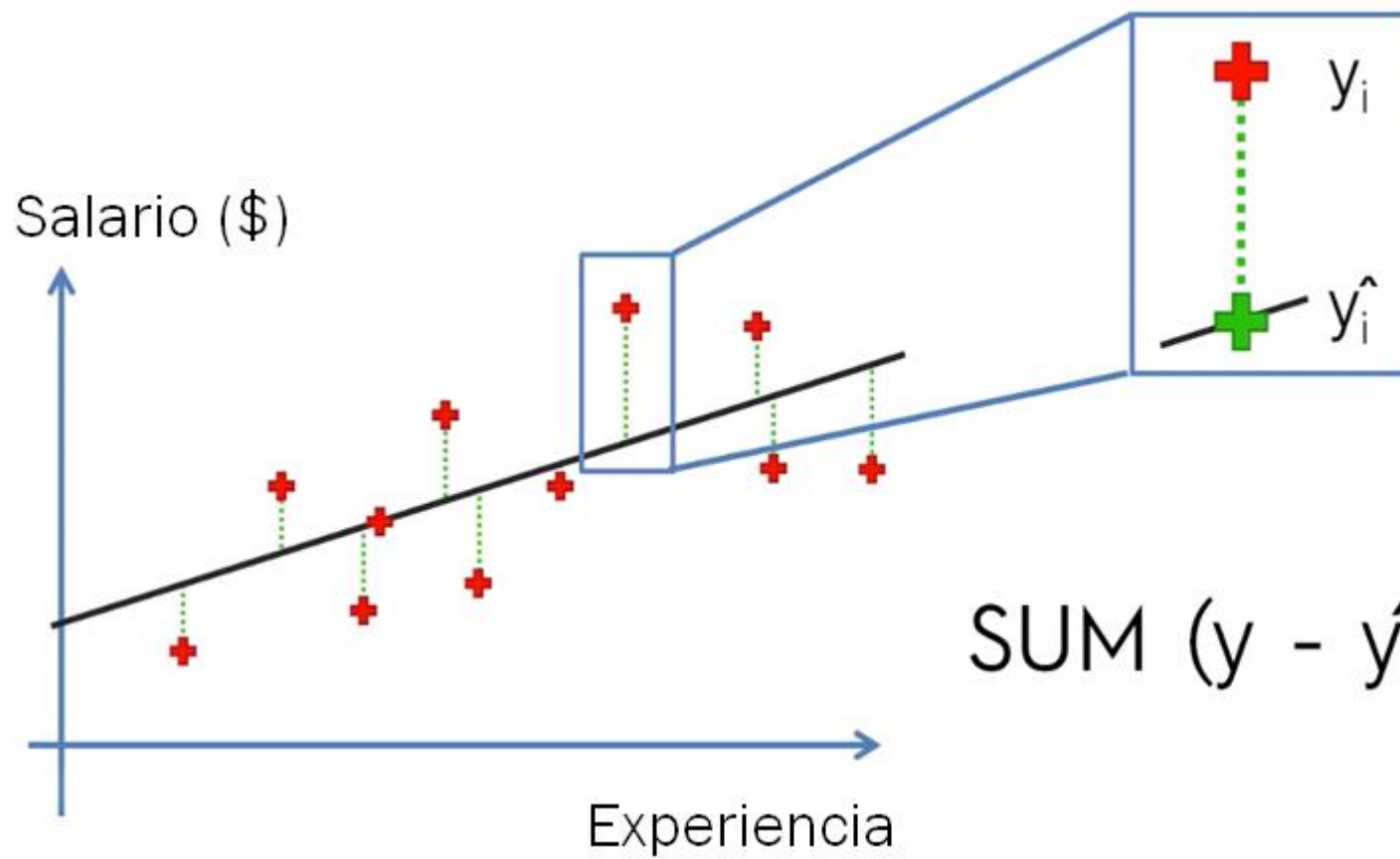
Independent variable (IV)



$$y = b_0 + b_1 * x$$




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




$$\text{SUM } (y - \hat{y})^2 \rightarrow \min$$

REGRESIÓN LINEAL SIMPLE EN PYTHON

- Pre-procesamiento de datos.
 - Ajustar nuestro regresor a nuestro train set.
 - Predecir los valores del test set.
 - Visualizar los resultados.
- 

PRE-PROCESAMIENTO DE DATOS

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
dataset = pd.read_csv('Salario_regresion.csv')
X = dataset.iloc[:, :1].values
y = dataset.iloc[:, 1].values
from sklearn.model_selection import train_test_split
X_train , X_test, y_train, y_test = train_test_split(X, y, test_size =
    1/3, train_size=2/3)
```



AJUSTE DE REGRESOR

- Importar la clase para regresión lineal simple.


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

- Crear un objeto de la clase con los parámetros necesarios.

```
regressor = LinearRegression()
```

- Aplicar los métodos de la clase a nuestros datos.

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



REALIZACIÓN DE PREDICCIONES

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

VISUALIZACIÓN DE RESULTADOS

```
plt.scatter(X_train, y_train, color = 'red')  
plt.plot(X_train, regresor.predict(X_train), color = 'blue')  
plt.title('Salario vs Experiencia (Training set)')  
plt.xlabel('Años de Experiencia')  
plt.ylabel('Salario')  
plt.show()
```