Regresión lineal y regresión logística

esmendoza@universidadean.edu.co
Estefanía Mendoza

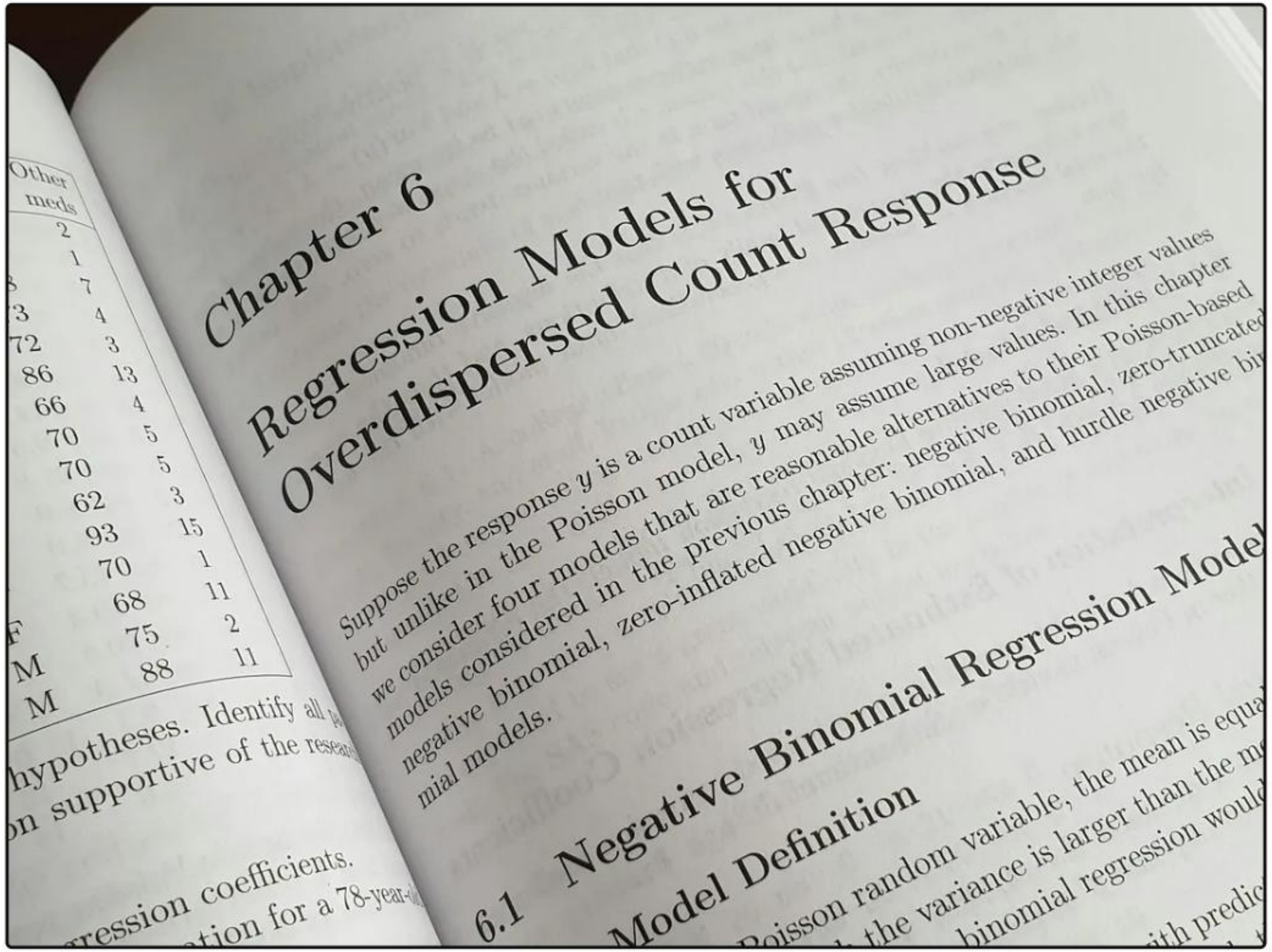


Foto de Enayet Raheem en Unsplash

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2$$

Sensible a los datos atípicos

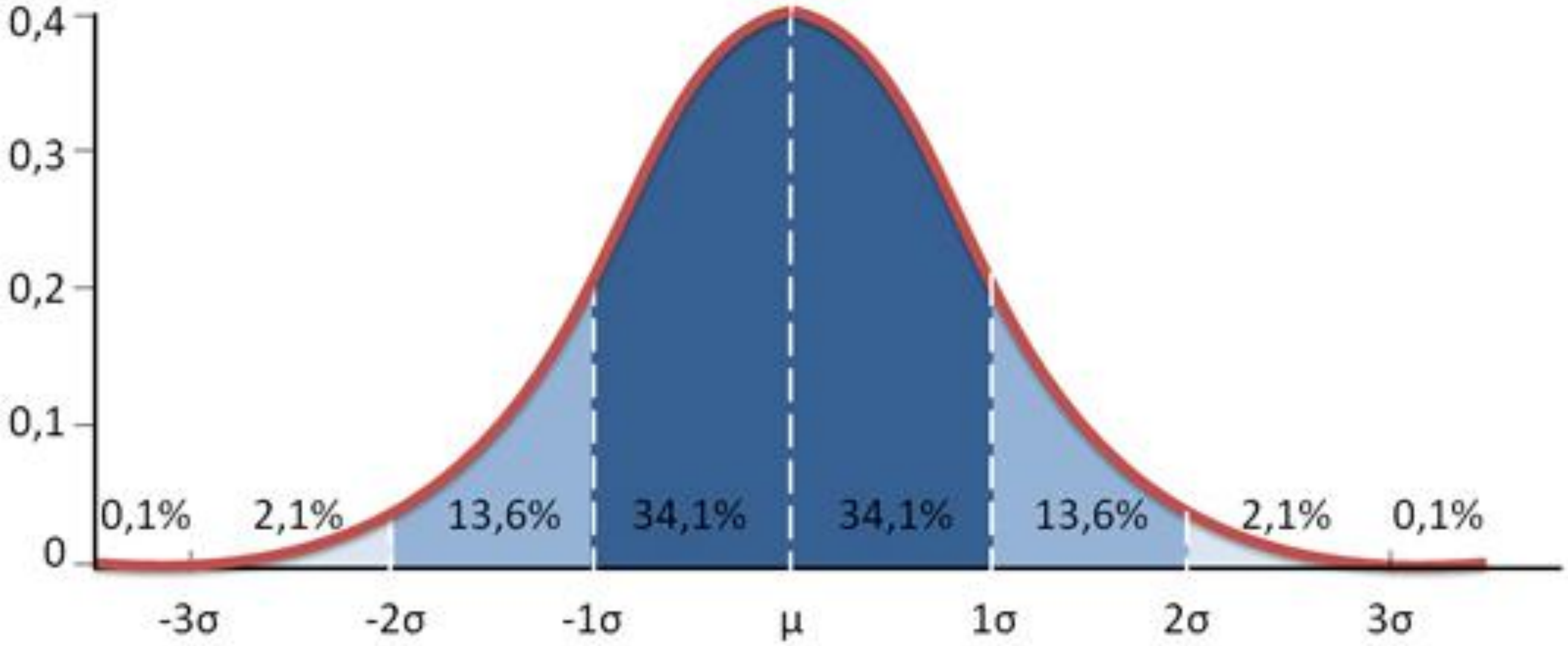
- σ^2 es la varianza de la población.
- N es el número total de datos en la población.
- x_i son los valores individuales.
- μ es la media de la población.

Interpretación: La varianza es una medida de cuánto varían los datos respecto a la media. Una varianza alta indica que los datos están muy dispersos, mientras que una varianza baja indica que están más concentrados alrededor de la media.

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$$

Error promedio de los datos

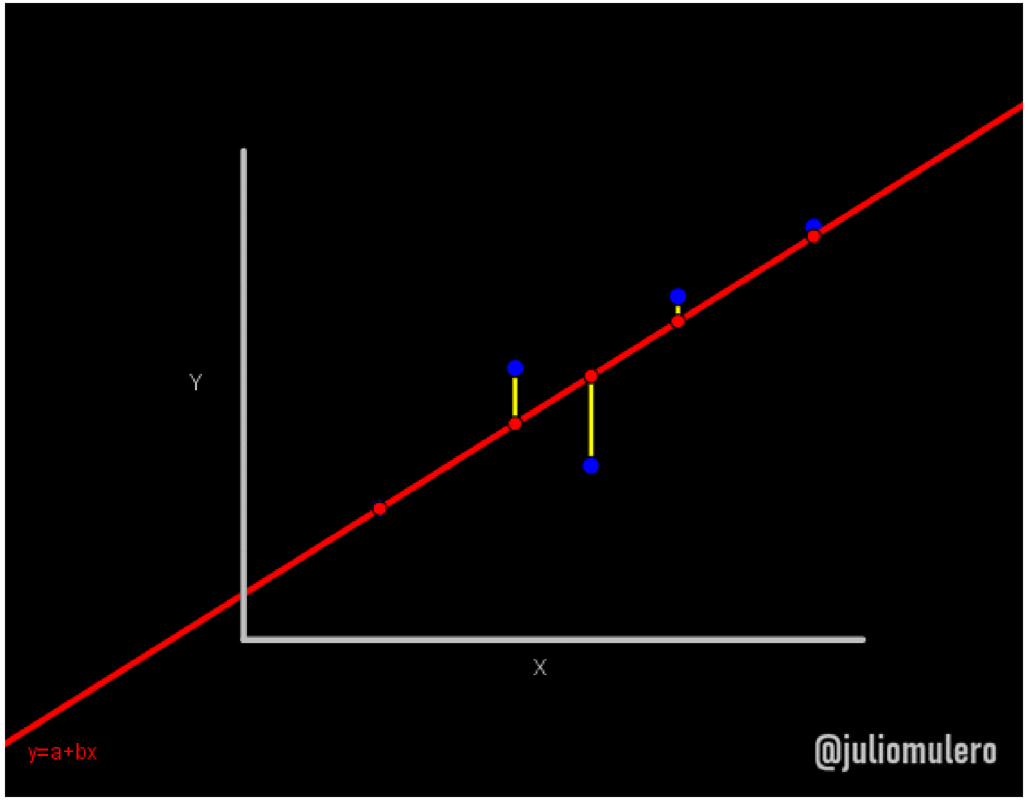
Interpretación: La desviación estándar proporciona una medida de dispersión en las mismas unidades que los datos originales. Esto la hace más intuitiva y fácil de interpretar en comparación con la varianza.


0,4
0,3
0,2
0,1
0
0,1%
2,1%
13,6%
34,1%
34,1%
13,6%
2,1%
0,1%
-3σ
-2σ
-1σ
μ
1σ
2σ
3σ

¿Qué es la regresión lineal?

La regresión lineal es una técnica de análisis de datos que predice el valor de datos desconocidos mediante el uso de otro valor de datos relacionado y conocido. Modela matemáticamente la variable desconocida o dependiente y la variable conocida o independiente **como una ecuación lineal**.

.



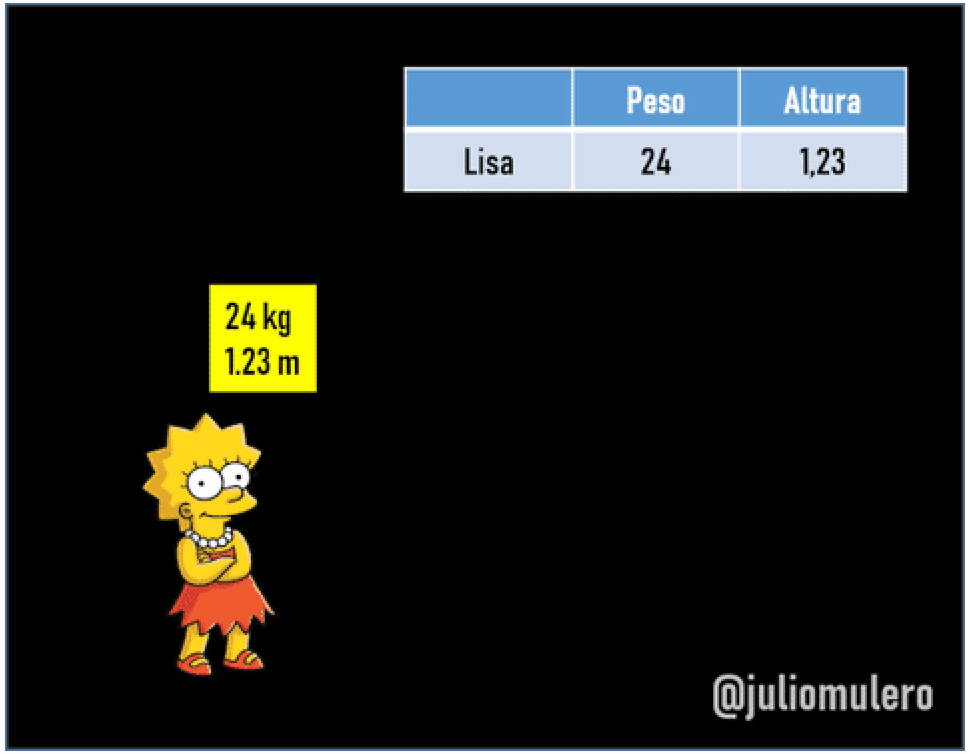
@aws

Imagina que quieres estudiar la relación entre la estatura y el peso.

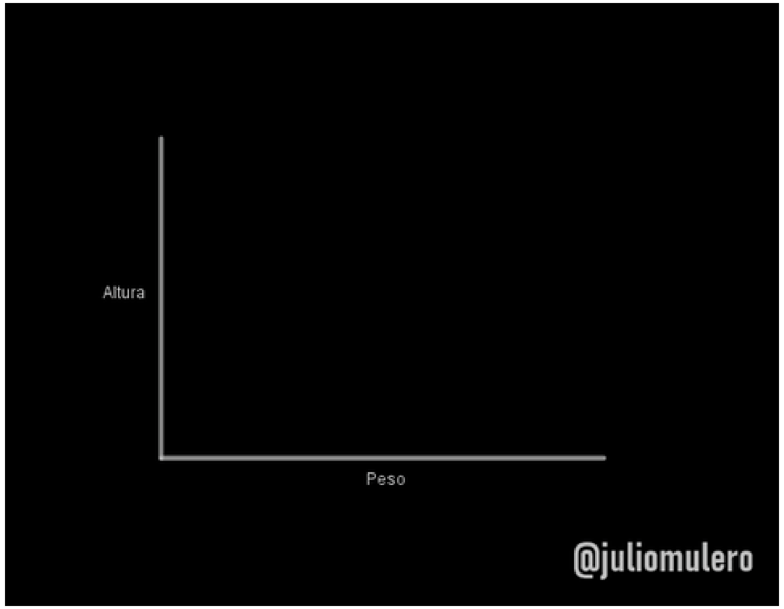
1.¿Si aumenta el peso, por ejemplo, aumenta también la estatura?

2.¿Se observa relación lineal entre ambas variables? ¿en qué medida?

(24kg,1.23m), (150kg,2.00m), (85kg,1.65m), (5,0kg.50m), (75kg, 1.85m).

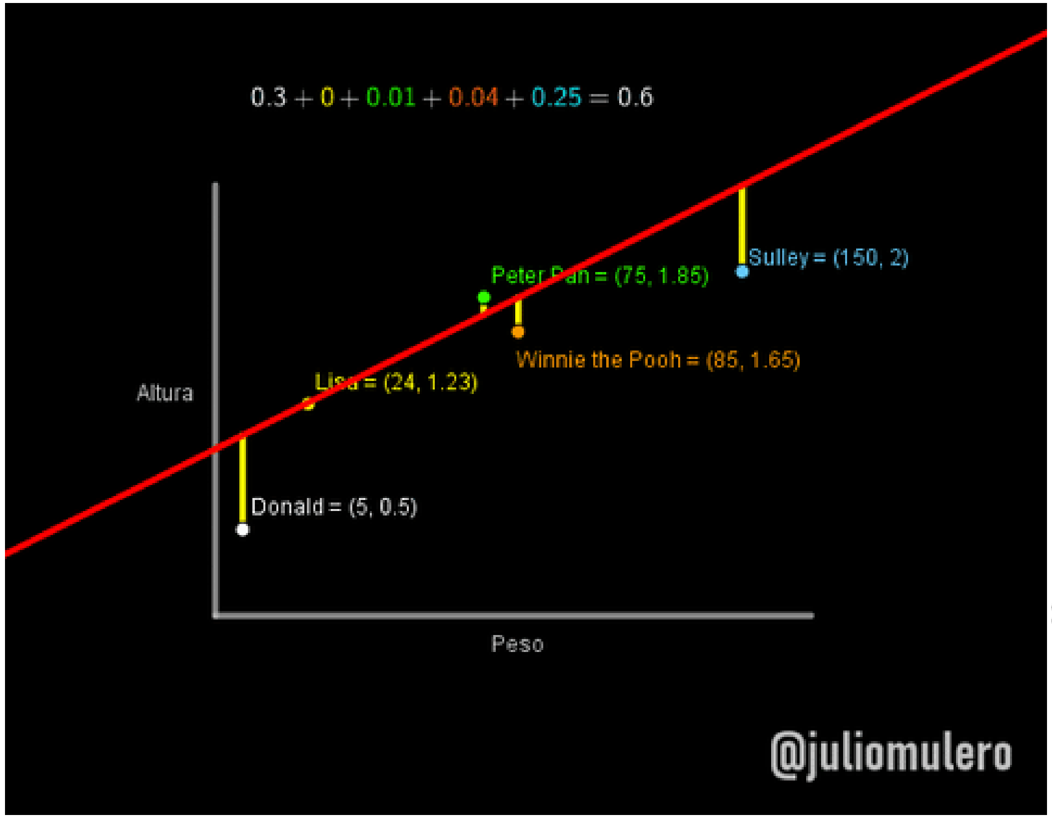


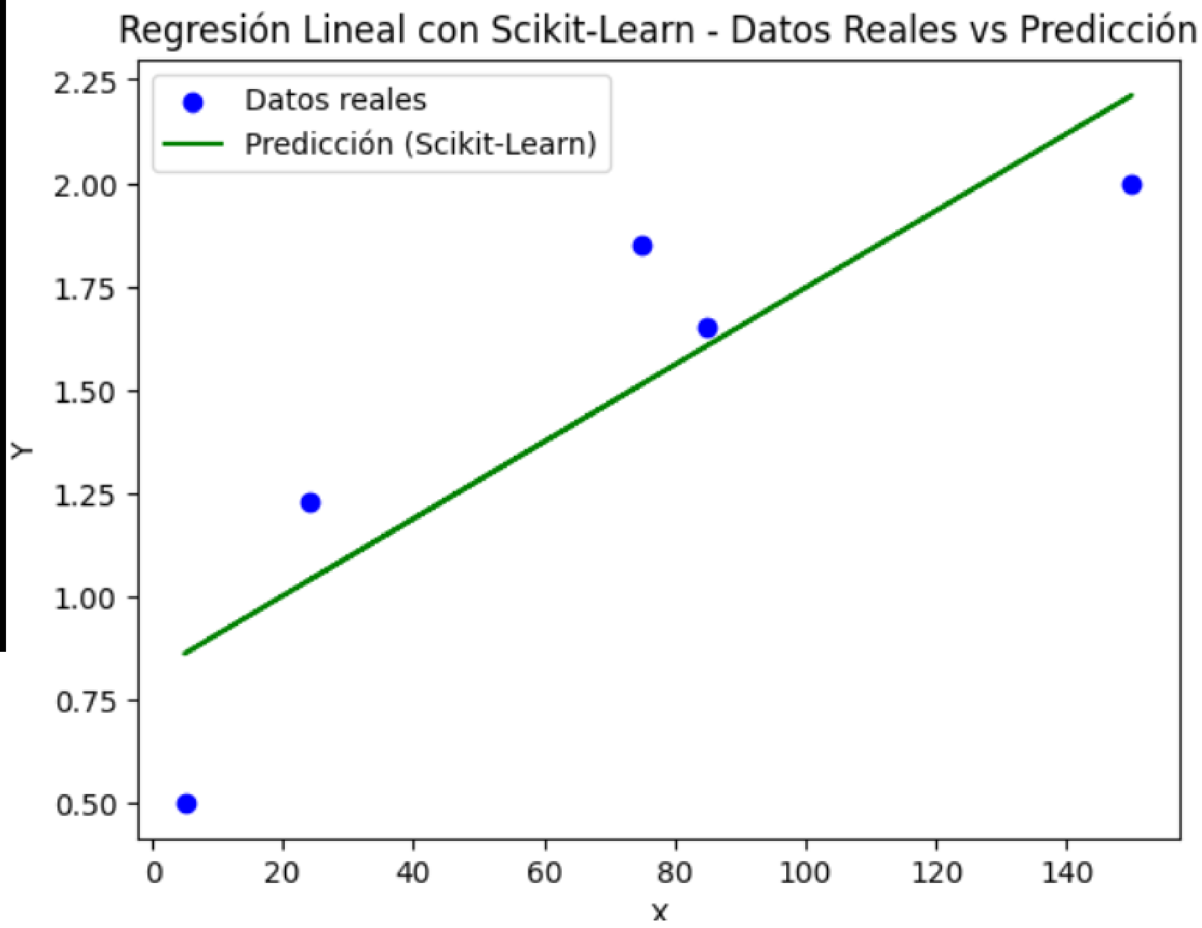
@elultimoversodefermat



Predictor

Respuesta $Y = mx + b$





Ecuación de la recta: Y = 0.0093X + 0.8142

Predicción para X = 35: Y = 1.1404

EL propósito principal es predecir el valor de la variable dependiente basándose en los valores de las variables independientes.

```python
# Calcular MAE
mae = mean_absolute_error(y, y_pred)
print(f"MAE: {mae}")

# Calcular MSE
mse = mean_squared_error(y, y_pred)
print(f"MSE: {mse}")

# Calcular RMSE
rmse = np.sqrt(mse)
print(f"RMSE: {rmse}")

# Calcular R²
r2 = r2_score(y, y_pred)
print(f"R²: {r2}")
```

ME … L

4. Coeficiente de …

Fórmula:

$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y} \ldots}{\sum_{i=1}^{n}(y_i - \bar{y} \ldots}$

Donde $\bar{y}$ es la media …

- **Descripción**: Mi… e es explicada por …res las variables inde…
- **Interpretación**: U… za de los datos, …n con mientras que un … anza.

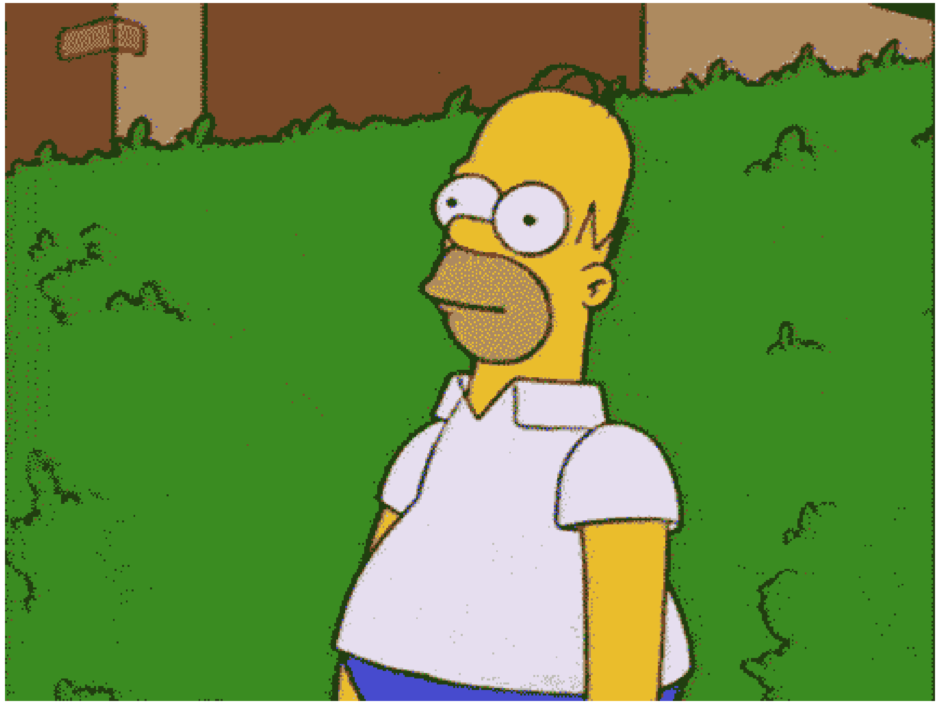
Información que Revelan estas Métricas

- **Ajuste del Modelo**: Métricas como el RMSE, MSE y MAE indican qué tan bien el modelo se ajusta a los datos. Un valor bajo en estas métricas sugiere un buen ajuste, pero también hay que tener en cuenta la posibilidad de sobreajuste (overfitting) si el modelo se ajusta demasiado bien a los datos de entrenamiento.
- **Robustez frente a Valores Atípicos**: El MAE y el RMSLE son más robustos frente a valores atípicos que el MSE, lo que es importante en conjuntos de datos donde los valores atípicos podrían sesgar el modelo.
- **Capacidad Explicativa**: El coeficiente R^2 muestra qué tan bien las variables independientes explican la variabilidad de la variable dependiente. Un alto R^2 generalmente sugiere un modelo más útil, pero debe interpretarse con precaución en modelos con muchas variables, donde el ajuste puede parecer artificialmente alto debido al sobreajuste.
- **Interpretabilidad**: El RMSE es interpretado en las mismas unidades que la variable dependiente, lo que facilita la comprensión directa de la magnitud de los errores de predicción.

Elegir una métrica o varias para optimizar el modelo con el que mejor este trabajando

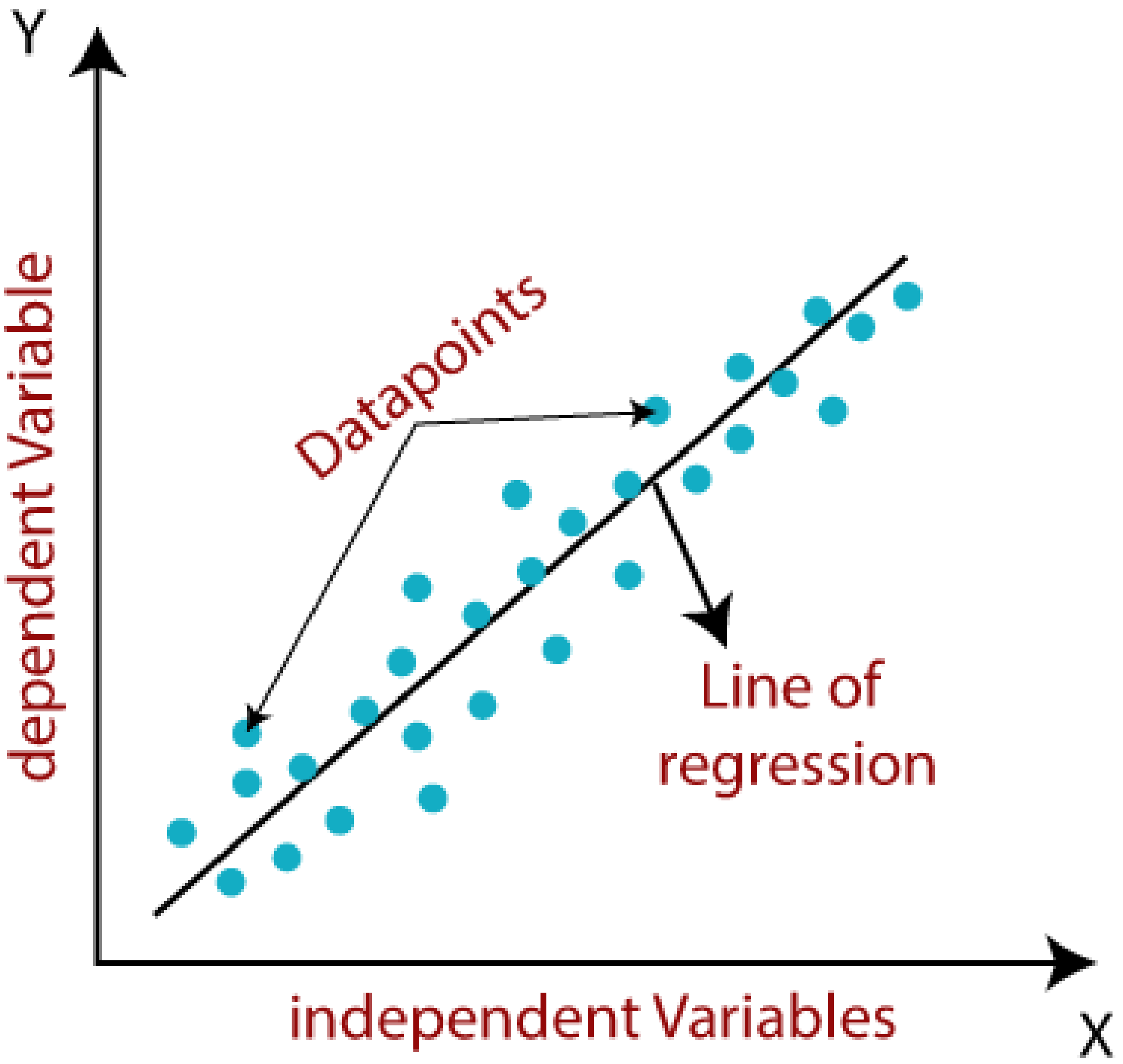
[Ejemplo en colab]

```
[3] import numpy as np
    import pandas as pd
    import matplotlib.pyplot as plt
    from sklearn.linear_model import LinearRegression
    from sklearn.metrics import mean_squared_error
```



¿ Que métrica seleccionar?

Variabl
quiere
Tambien
Variable
de respu
predicha
endoge
etc..



ible

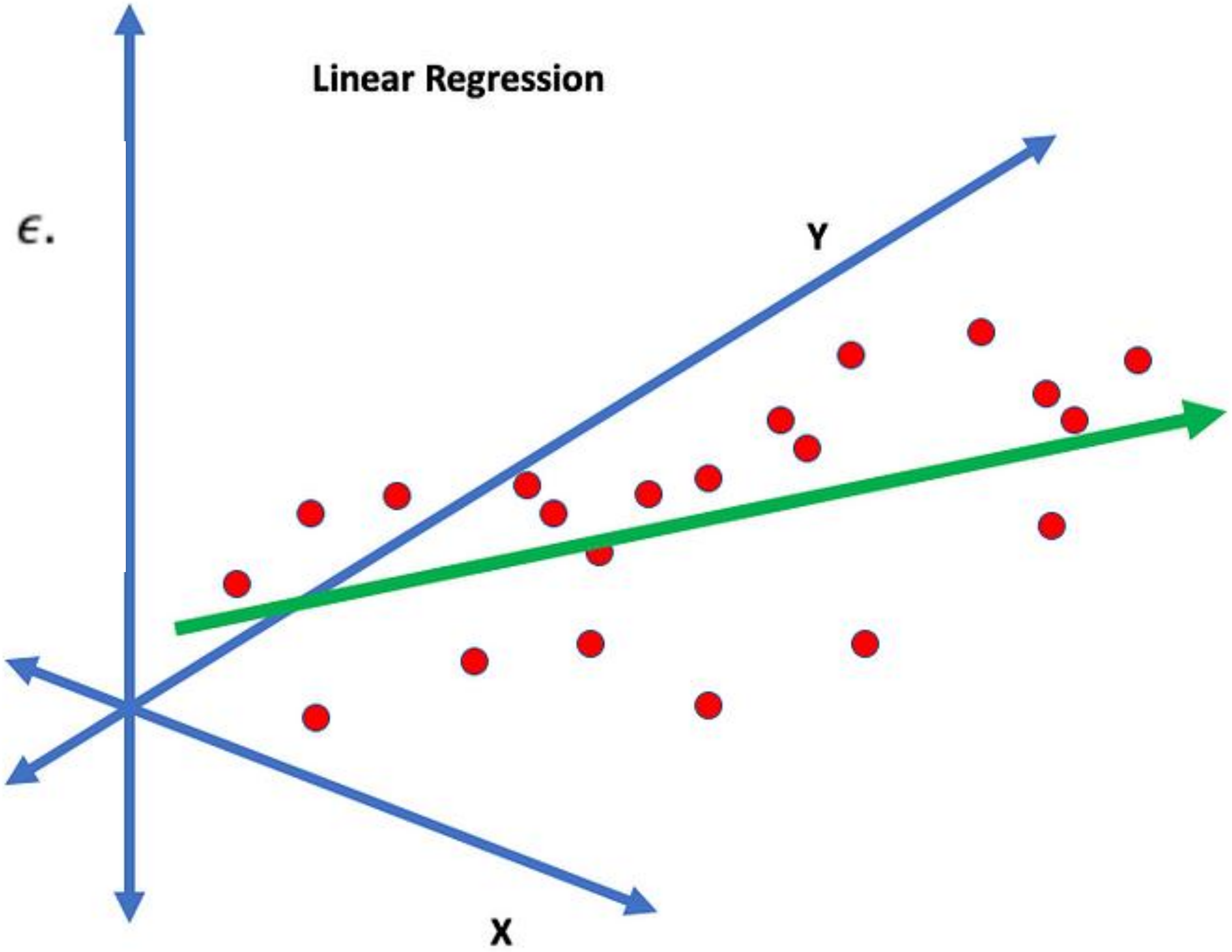
s residuos

a

La regresión lineal múltiple (MLR) es un método estadístico que se utiliza para modelar la relación entre dos o más variables independientes y una variable dependiente. En el contexto del aprendizaje automático, es un algoritmo de aprendizaje supervisado que puede predecir un resultado continuo.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \epsilon.$$

$$y = \beta_0 + \sum_{i=1} \beta_i x_i + \epsilon_i.$$



Modelo de Predicción con Múltiples Variables:

- El modelo se representa mediante una suma ponderada de varias variables de entrada.
- La fórmula es:

$$f_{w,b}(\mathbf{x}) = w_0x_0 + w_1x_1 + \cdots + w_{n-1}x_{n-1} + b$$

- Aquí, $w_0, w_1, \ldots, w_{n-1}$ son los pesos asociados a cada variable de entrada $x_0, x_1, \ldots, x_{n-1}$, y b es el sesgo

$$\boxed{f_{w,b}(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b}$$

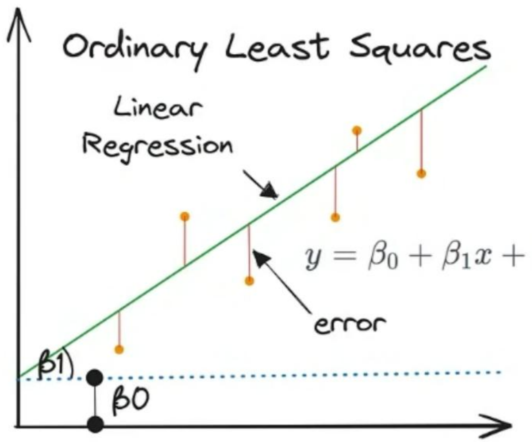
- Donde $\mathbf{w}$ y $\mathbf{x}$ son vectores y el operador $\cdot$ representa el producto punto (dot product).

El coeficiente de mayor valor es el mas relevante en los precios, que tanto incide una variable en una predicción

Cada valor de **x** tiene una incidencia en el modelo y **w** cuanto pesa la variable, para encontrar la ecuación que me haga la mejor predicción posible y para esto el modelo tiene que irse actualizando para ir encontrado los coeficientes que debe tener cada variable para que la predicción sea muy buena y minimizar el error que es la **diferencia entre REALIDAD Y PREDICCION** .

Función de Costo $J(w,b)$:

- La función de costo se denota como $J(w,b)$ y mide el promedio del error cuadrático entre las predicciones del modelo y los valores reales en el conjunto de entrenamiento.



Aprox método

Datos reales

$$J(w,b) = \frac{1}{2m} \sum_{i=0}^{m-1} \left(f_{w,b}(x^{(i)}) - y^{(i)} \right)^2$$

$$f_{w,b}(x^{(i)}) = \mathbf{w} \cdot \mathbf{x}^{(i)} + b$$

¿PUEDO MEJORAR EL VALOR DE LA FUNCION DE COSTO?

Error diferencia entre la predicción del modelo y la realidad

4. **Coeficiente de Determinación (R^2 o R-cuadrado)**

Fórmula:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

Donde $\bar{y}$ es la media de los valores reales.

- **Descripción**: Mide la proporción de la varianza en la variable dependiente que es explicada por las variables independientes en el modelo.
- **Interpretación**: Un R^2 cercano a 1 indica que el modelo explica bien la varianza de los datos, mientras que un R^2 cercano a 0 sugiere que el modelo no explica bien la varianza.

3. **Error Absoluto Medio (Mean Absolute Error, MAE)**

Fórmula:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

- **Descripción**: Calcula la media de las diferencias absolutas entre los valores reales y los valores predichos.
- **Interpretación**: El MAE es una métrica más robusta frente a valores atípicos en comparación con el MSE. Un MAE bajo indica un modelo con un ajuste preciso.

1. **Error Cuadrático Medio (Mean Squared Error, MSE)**

Fórmula:

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

- **Descripción**: Mide la magnitud promedio de los errores al cuadrado entre los valores reales (y los valores predichos ($\hat{y}_i$).
- **Interpretación**: Un MSE bajo indica que las predicciones del modelo están, en promedio, cer de los valores reales. Sin embargo, debido a que los errores se elevan al cuadrado, los valore atípicos pueden tener un gran impacto en el MSE.

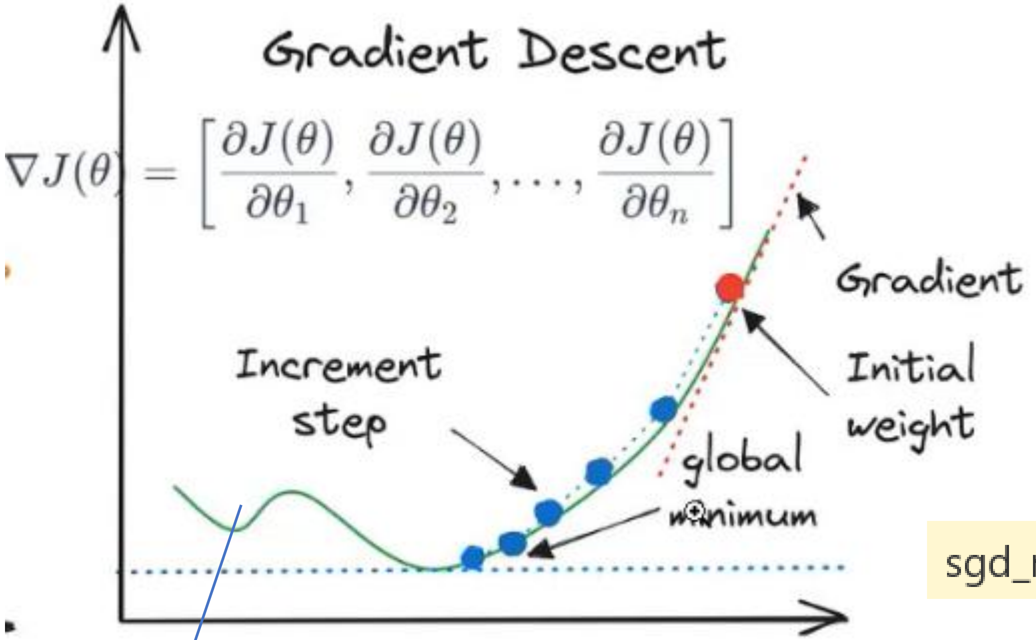
2. **Raíz del Error Cuadrático Medio (Root Mean Squared Error, RMSE)**

Fórmula:

$$\mathrm{RMSE} = \sqrt{\mathrm{MSE}}$$

- **Descripción**: Es la raíz cuadrada del MSE. Representa la desviación estándar de los errores de predicción.
- **Interpretación**: El RMSE tiene la misma unidad que la variable de salida, lo que facilita la interpretación. Un RMSE más bajo indica un mejor ajuste del modelo.

El gradiente descendente es un algoritmo de optimización utilizado para minimizar funciones de costo en el contexto del aprendizaje automático y el análisis de datos.



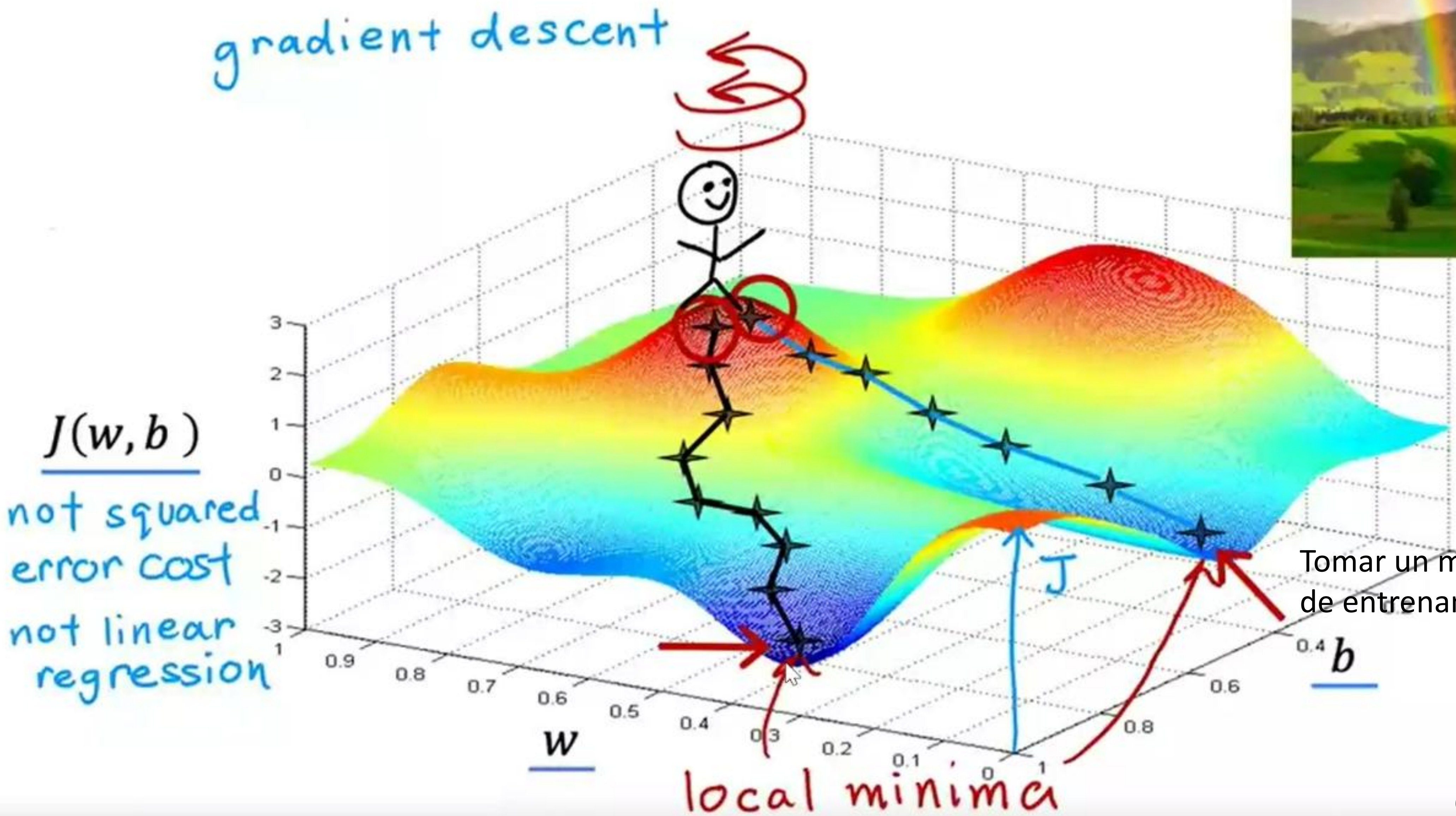
Mínimo local

En scikit-learn, la **tasa de aprendizaje** es un hiperparámetro crucial para los algoritmos de optimización basados en gradientes, como los que se utilizan en varios modelos de aprendizaje automático. Determina el tamaño del paso en cada iteración al moverse hacia un mínimo de una función de pérdida.

```
sgd_reg = SGDRegressor(loss='squared_error', max_iter=1000, tol=1e-3)
```

Concepto Básico

El objetivo del gradiente descendente es encontrar los valores de los parámetros del modelo que minimizan la función de costo (o error). La función de costo mide cuán bien el modelo predice los datos de entrenamiento.



Tomar un mal dato de entrenamiento

@stanford

Tasa de Aprendizaje en Diferentes Modelos

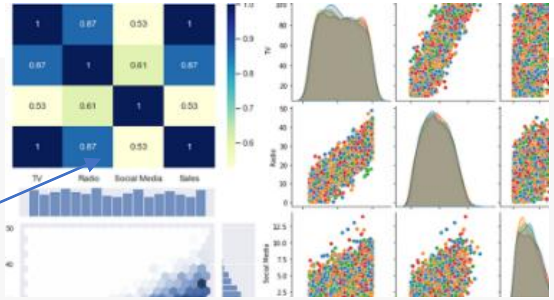
1. Descenso de Gradiente Estocástico (SGD)

   Para modelos lineales y máquinas de soporte vectorial, scikit-learn proporciona `SGDClassifier` y `SGDRegressor`, donde se puede especificar la tasa de aprendizaje utilizando el parámetro `eta0` junto con un esquema de tasa de aprendizaje especificado por el parámetro `learning_rate`.

python Copiar código

```python
from sklearn.linear_model import SGDClassifier

sgd_clf = SGDClassifier(learning_rate='constant', eta0=0.01)
sgd_clf.fit(X_train, y_train)
```



```python
# importing modules and packages
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score, explained_
from sklearn import preprocessing
```

NORMALIZAR

```
X_tra
```

```python
model = LinearRegression()
```

```python
[ ] model.fit(X_train,y_train)
```

```
▾ LinearRegression
LinearRegression()
```

```python
[ ] # model evaluation
    print(
      'mean_squared_error : ', mean_squared_error(y_test, predictions))
    print(
      'mean_absolute_error : ', mean_absolute_error(y_test, predictions))
```

```
mean_squared_error :  0.3227216962941032
mean_absolute_error :  0.3550106060261852
```

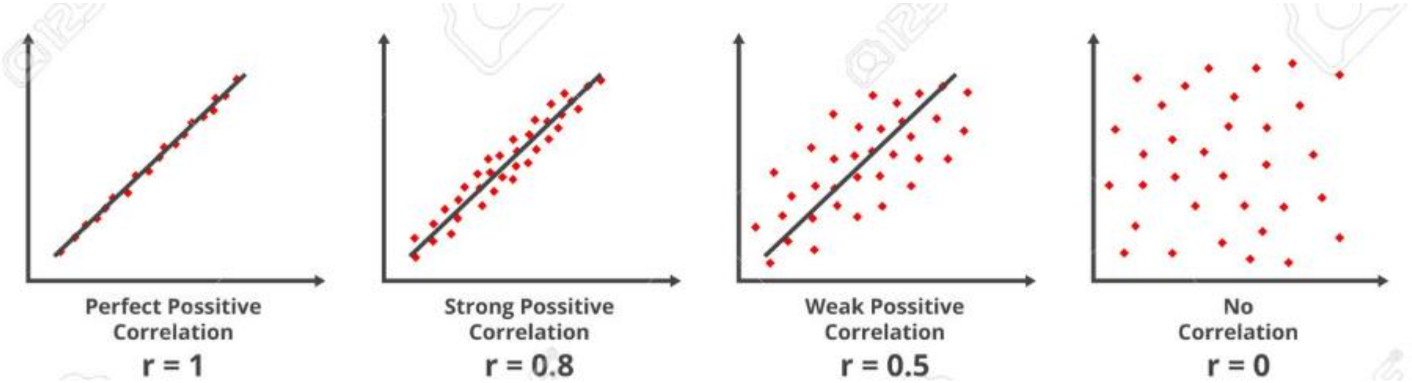
En una sola columna

Fit minimizar la función de costo, se ajusta muchas veces

Y test y predicción es lo que el modelo predice

Tarea : Consultar como y por que normalizar variables en bases de datos.

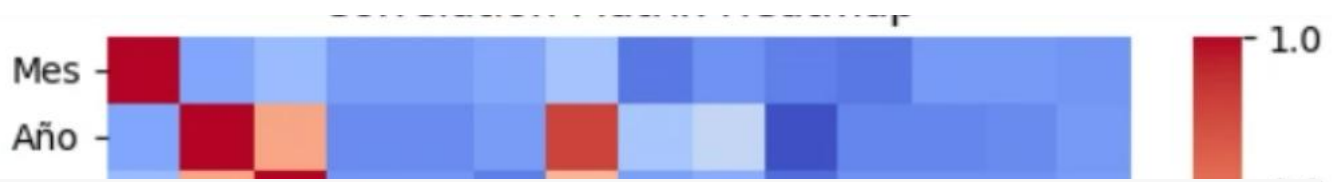
CORRELACION



¿ Que es un buen modelo ?

1 Variables con Valores de X significativos
2. Variables Poco correlacionadas (0.7)
3. Variables escaladas
4. Modelo con bajo sesgo

Importante correlación (0.9)

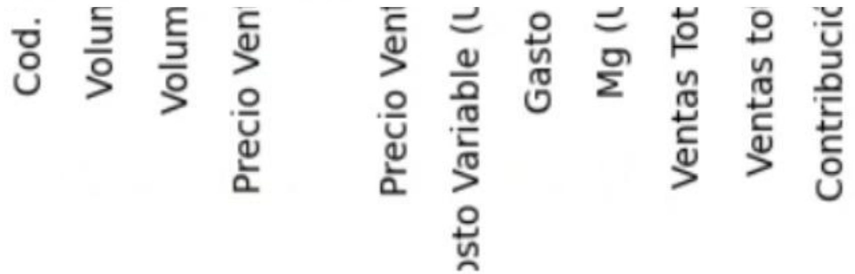


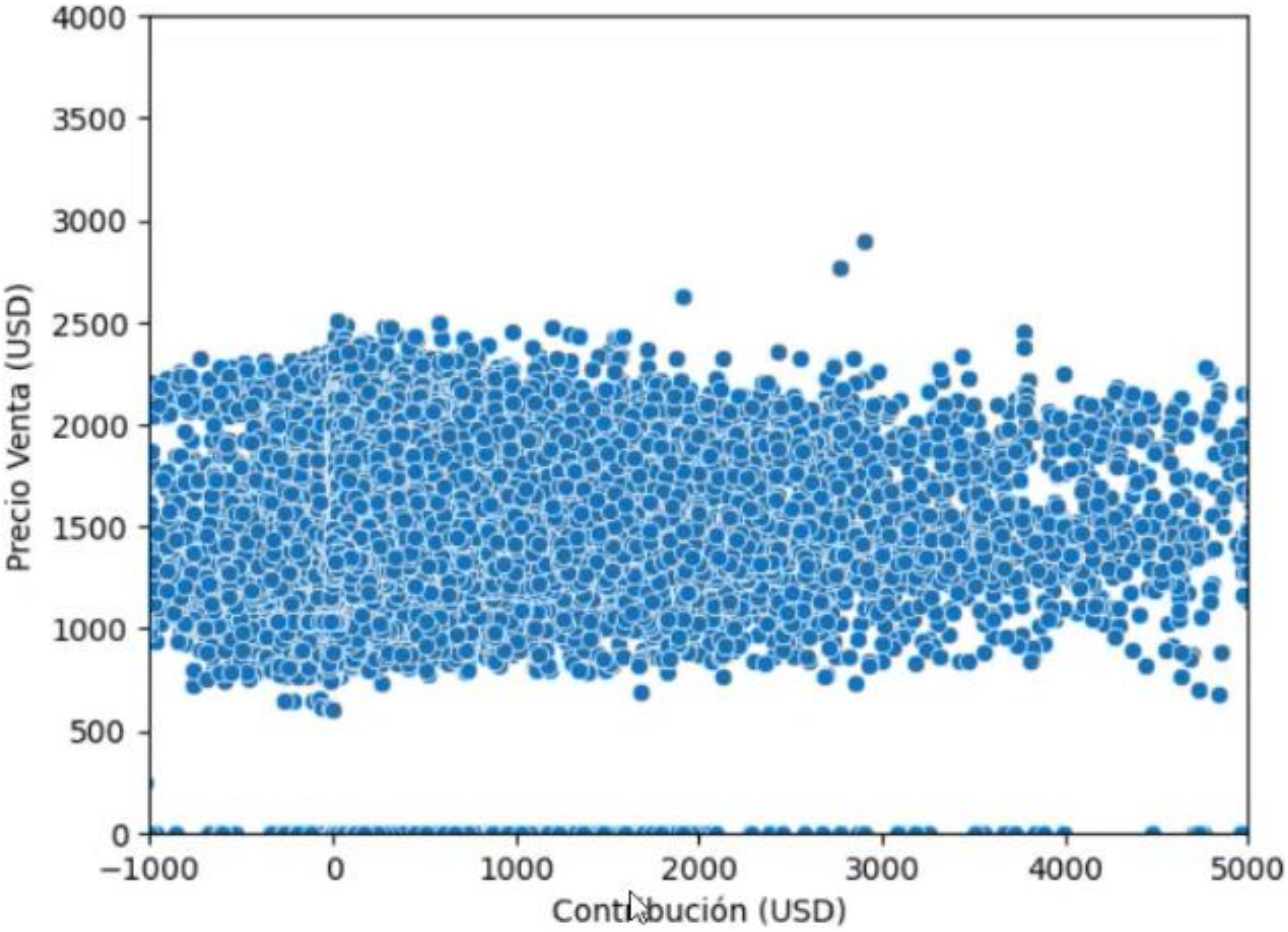

```python
import numpy as np
cm = np.corrcoef(df[cols].values.T)
sns.set(font_scale=1.5)
sns.heatmap(cm,cbar=True,annot=True, yticklabels=cols, xticklabels=cols)
```

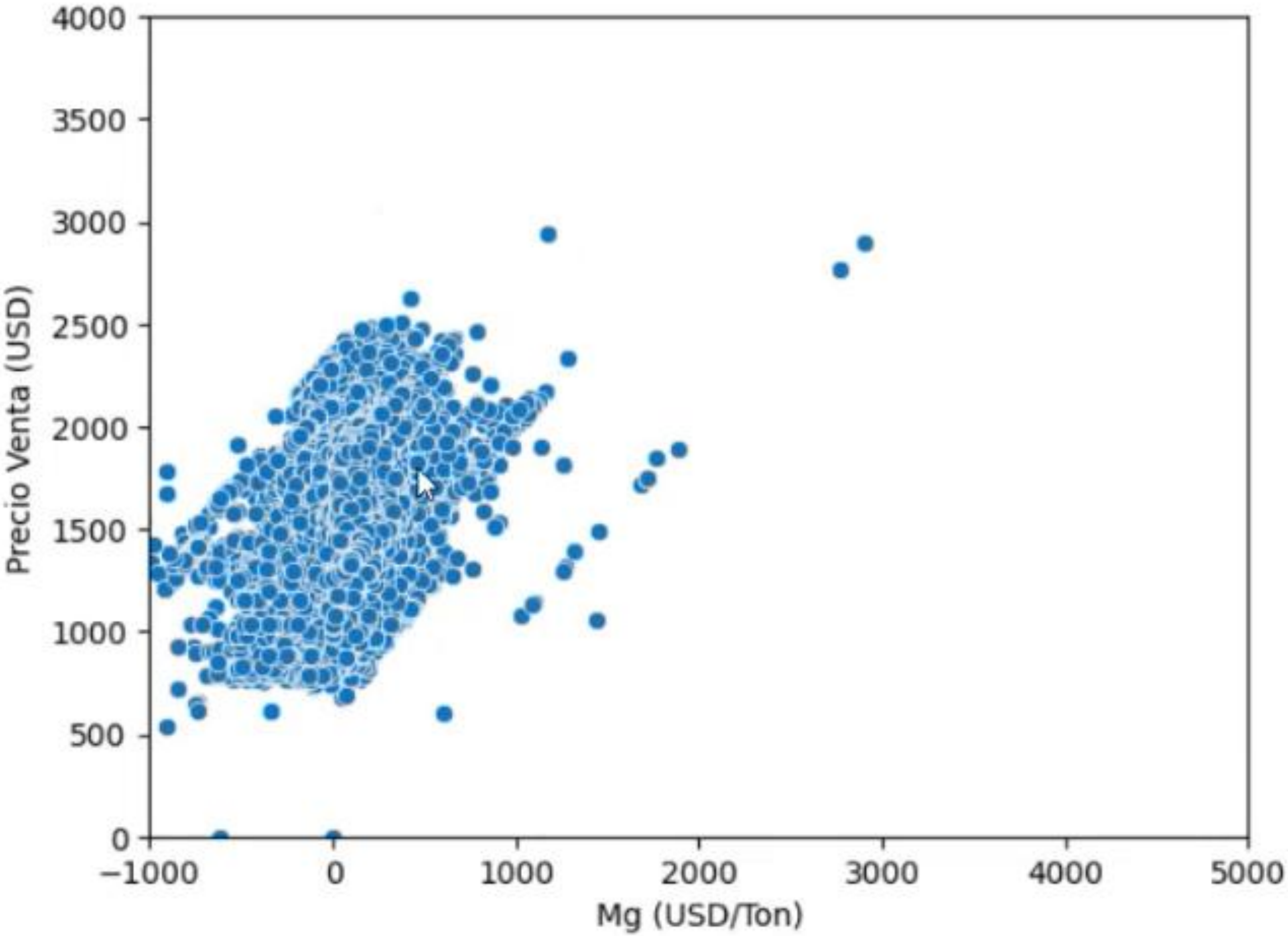
<matplotlib.axes._subplots.AxesSubplot at 0x7fbf3ae92b90>


```python
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwa

# Agregar títulos y etiquetas
plt.title('Mapa de Calor de la Matriz de Correlación')
plt.show()
```




Precio Venta (USD)
4000
3500
3000
2500
2000
1500
1000
500
0
−1000
0
1000
2000
3000
4000
5000


Precio Venta (USD)
4000
3500
3000
2500
2000
1500
1000
500
0
−1000
0
1000
2000
3000
4000
5000
Mg (USD/Ton)

Correlación de Pearson

La **correlación de Pearson** es una medida estadística que evalúa la fuerza y la dirección de la relación lineal entre dos variables cuantitativas. Se denota comúnmente como r o ρ.

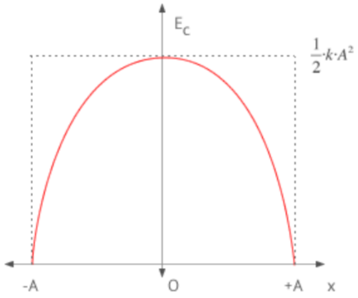
Fórmula de la Correlación de Pearson

La fórmula para calcular la correlación de Pearson entre dos variables X y Y es:

$$r = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2 \sum_{i=1}^{n}(Y_i - \bar{Y})^2}}$$

Limitaciones

- **Relación no lineal**: Pearson solo mide la correlación lineal. Si la relación entre las variables es no lineal, el coeficiente de Pearson puede no reflejar adecuadamente esa relación.



Ejemplos de Relaciones No Lineales

Algunos ejemplos de relaciones no lineales incluyen:

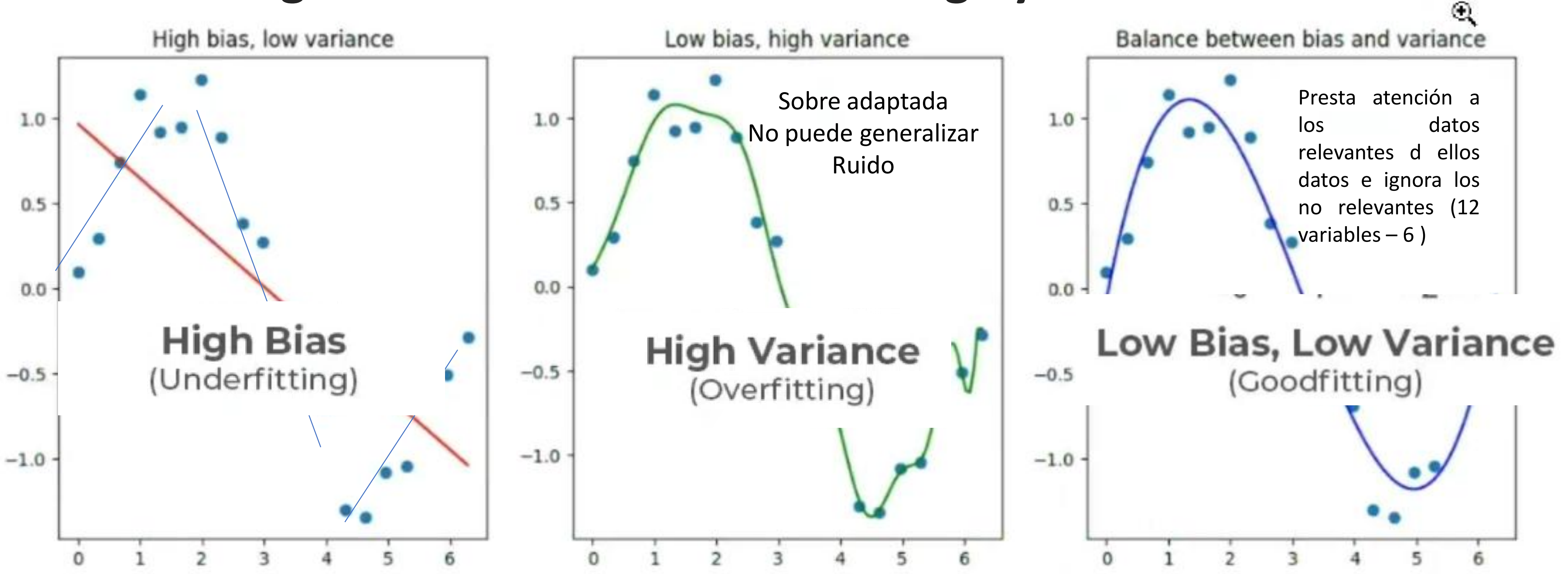
1. **Relación Cuadrática:**
   - Ejemplo: La relación entre la velocidad de un objeto y su energía cinética ($E = \frac{1}{2}mv^2$).
   - Gráficamente, esta relación se muestra como una parábola.

```
# Calcular correlación de Spearman
spearman_corr, _ = spearmanr(X, Y)
print(f"Correlación de Spearman: {spearman_corr}")

# Calcular correlación de Kendall
kendall_corr, _ = kendalltau(X, Y)
print(f"Correlación de Kendall: {kendall_corr}")
```

¿Que tan erróneo es un modelo?

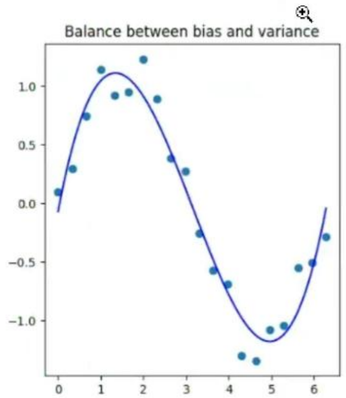
Diagnóstico del balance entre sesgo y varianza.

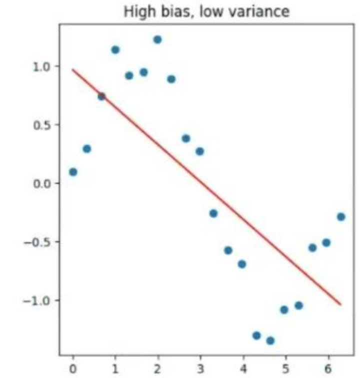


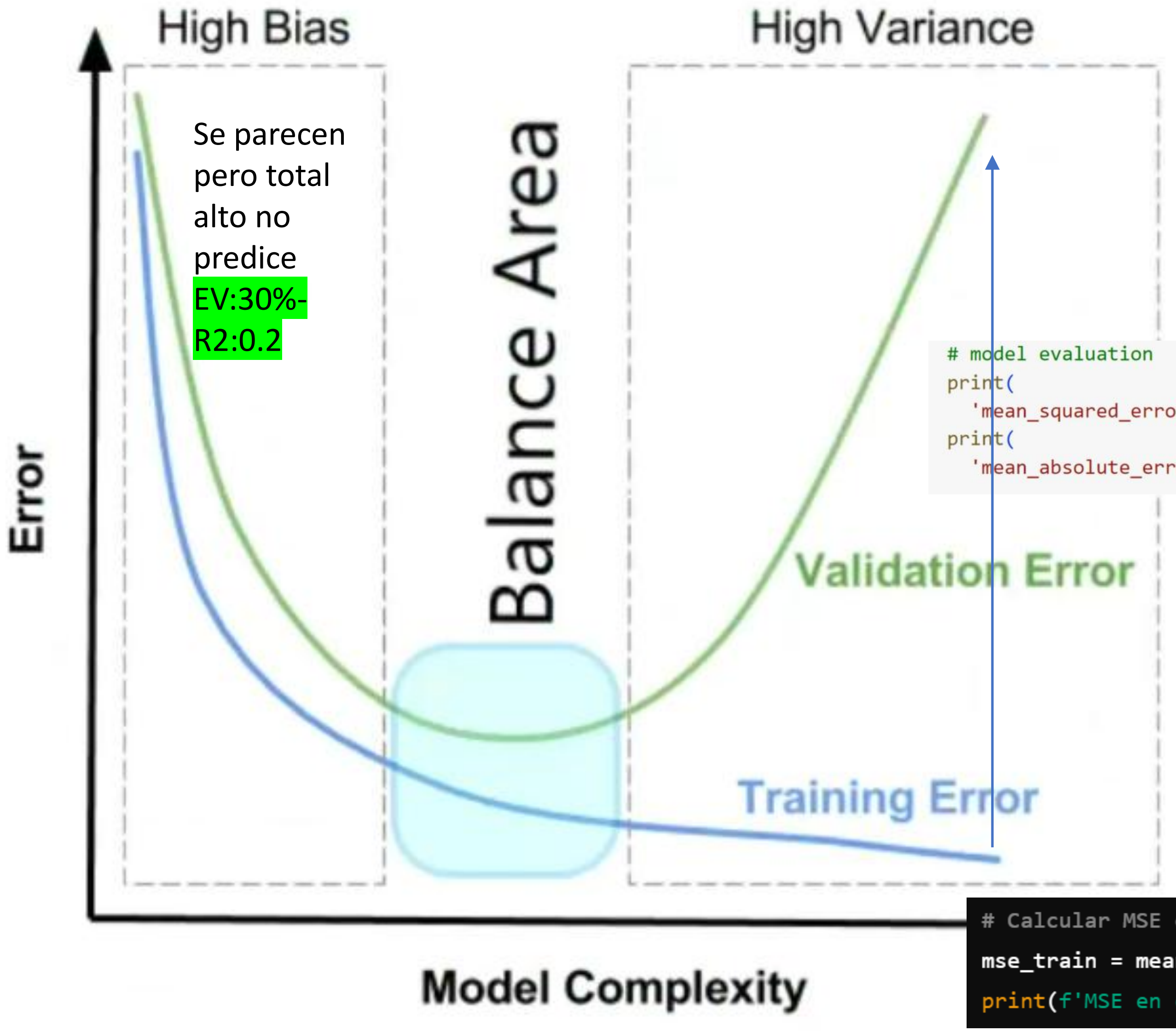
Un modelo tiene sesgo máximo. Cuándo a pesar de haber trazado una línea recta que tienen mínima distancia con todos los puntos. La línea es recta, o sea, es no se adaptó a la distribución de los datos, sino que solo llegó como a un promedio, pero es un modelo demasiado simple.

Que tanto el modelo sesga la varianza y es qué tanto el modelo ignora la verdadera varianza. Sobre simplifica la ecuación final y gráficamente

Diagnóstico del balance entre sesgo y varianza.







Se parecen pero total alto no predice
EV:30%-
R2:0.2

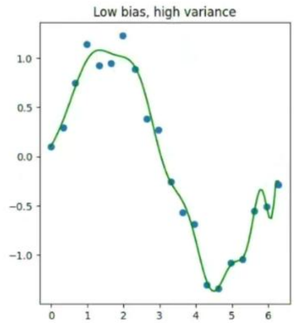


Image by author.

Alta varianza
Captura el ruido
Momentos específicos

```
# model evaluation
print(
  'mean_squared_error : ', mean_squared_error(y_test, predictions))
print(
  'mean_absolute_error : ', mean_absolute_error(y_test, predictions))
```

Métricas (Los datos de test 20-30%)

```
# Calcular MSE en los datos de entrenamiento
mse_train = mean_squared_error(y_train, y_train_pred)
print(f'MSE en el conjunto de entrenamiento: {mse_train}')
```

CRITERIO:Parecidos y error total bajo 3+3=6

Que pasa si

Cuidado con algo que puede pasar muy fácil en machine Learning, en regresión lineal y en todo el machine Learning, que es cuando el error es tan bajito y el R (0.05) dos es tan alto que usted dice venga, pero no se equivoca casi. Error promedio (0.8)

Lo expongo a datos nuevos NO FUNCIONA

Criterios de un modelo ideal.

1.Un buen balance entre sesgo y varianza.

2. Un modelo que presta atención a cambios relevantes en los datos e ignora los cambios no relevantes en los datos

Relaciones irrelevantes, son variaciones locales de los datos que se deban al azar al ruido (Un momento muy específico)

3. Un modelo que el último y más importante generaliza bien con datos que no conoce o hace buenas predicciones para datos que no conoce

El Polinomio sea lo suficientemente complejo para predecir los datos, pero no tan complejo que ya estés sobre adaptado a los datos y con datos nuevos no aprendan nada.

Deseo que el modelo no aprenda cosas demasiado específicas, no quiero que capture ruido, ni que capture cosas que solo pasan en ese punto, sino que el modelo generalice los nuevos datos.

El haya capturado relaciones entre las variables que sean extrapolables a otros periodos de tiempo.

"Everything should be made as simple as possible, but not simpler."

Albert Einstein