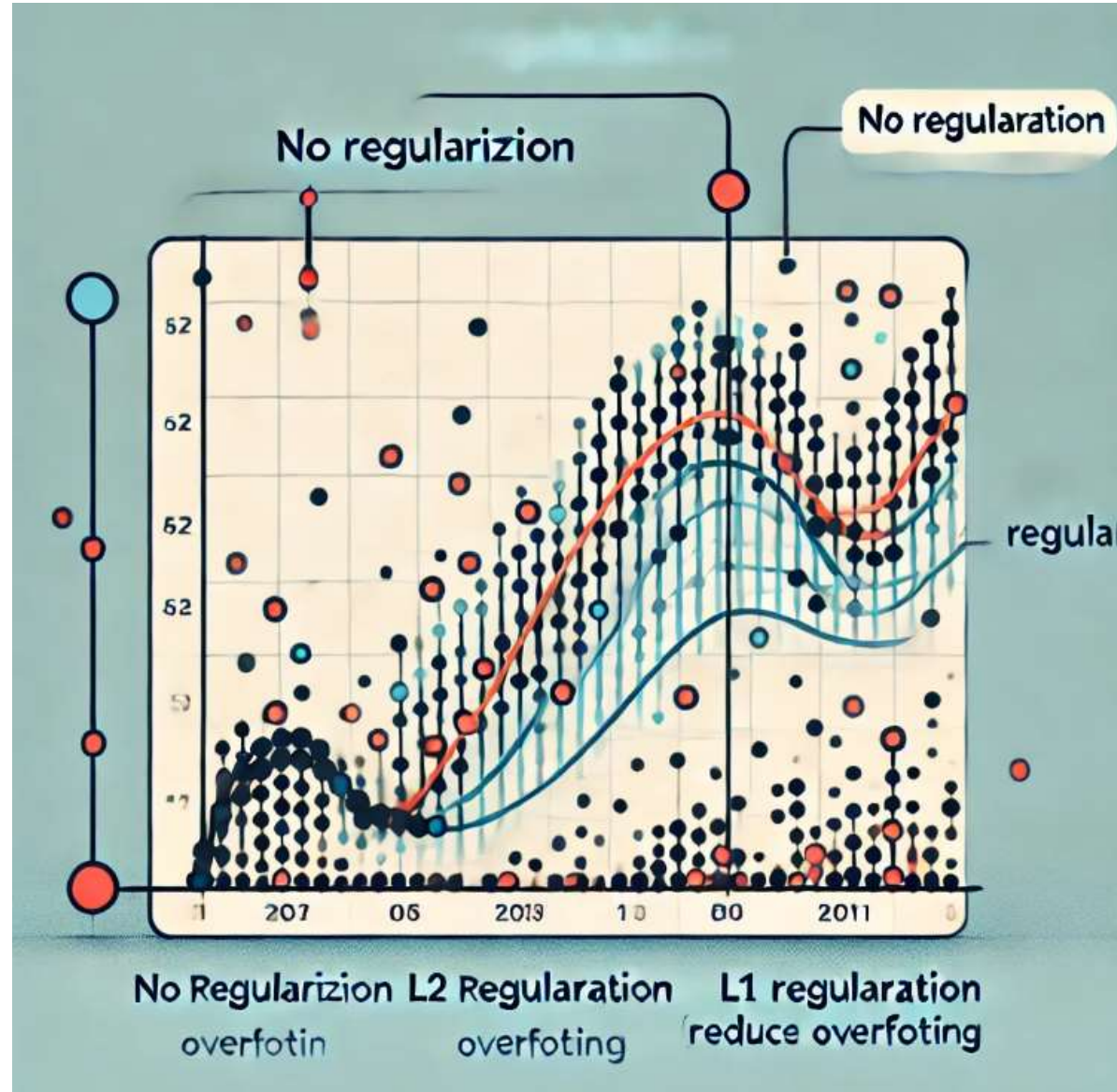
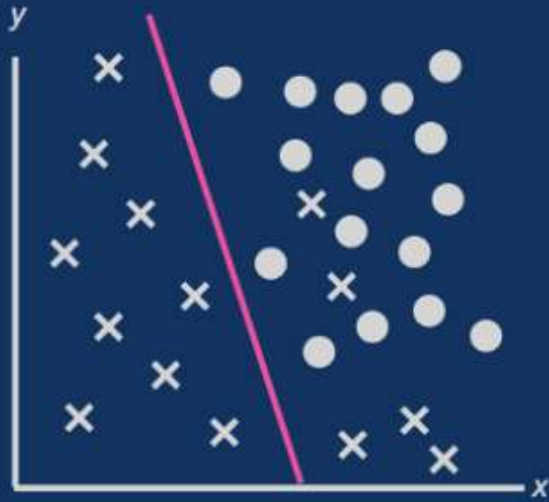


# REGULARIZACION

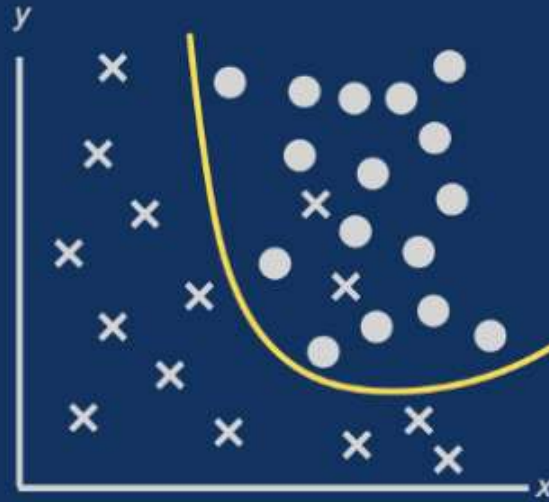


# Regularization

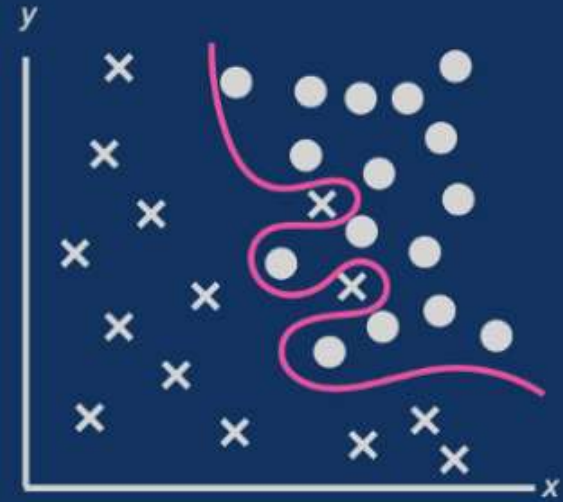
Underfitting



A good fit



Overfitting



*Regularization helps prevent overfitting*

## Fórmula del Error Cuadrático Medio (MSE)

La fórmula general para el error cuadrático medio (MSE) es:

$$E(\epsilon) = E((\hat{Y} - Y)^2)$$

Donde:

- $\hat{Y}$  es la predicción del modelo.
- $Y$  es el valor real.

**Varianza :**

Representa la cantidad en la que la predicción del modelo cambiaría si se usaran diferentes datos de entrenamiento (RUIDO).

### Descomposición del MSE

El MSE se puede descomponer en tres componentes: el sesgo al cuadrado, la varianza y el error irreducible ( $\sigma^2$ ).

$$E(\epsilon) = E((\hat{Y} - Y)^2) = (E(\hat{Y}) - Y)^2 + E((\hat{Y} - E(\hat{Y}))^2) + \sigma^2$$

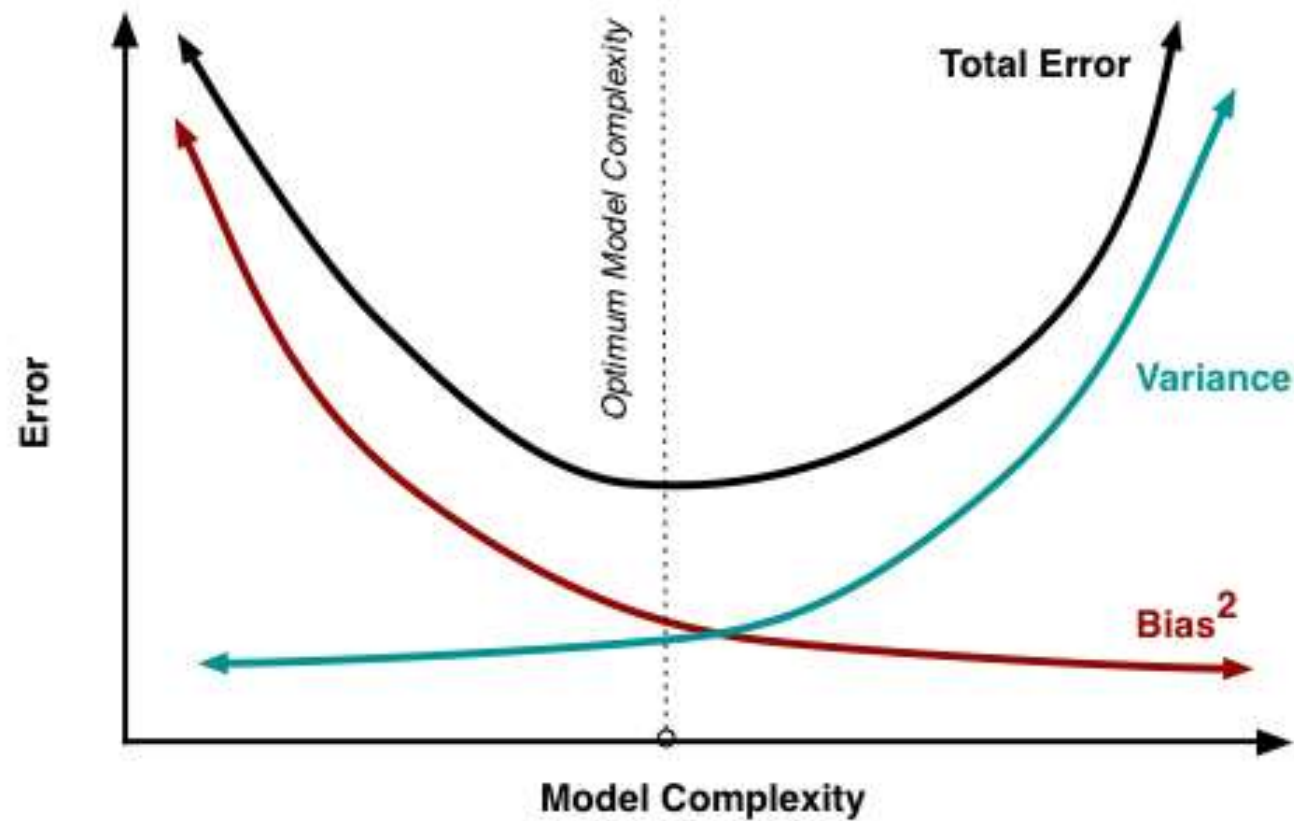
**Sesgo<sup>2</sup> :**

Representa el error debido a suposiciones erróneas en el modelo. Un modelo con alto sesgo tiende a ser demasiado simple y no captura adecuadamente la relación entre las características y la variable objetivo

**Error Irreducible**

Es el ruido inherente en cualquier problema de modelado de datos, debido a factores no considerados en el modelo ( no se reduce )

## Reducir la varianza para aumentar un poco el sesgo

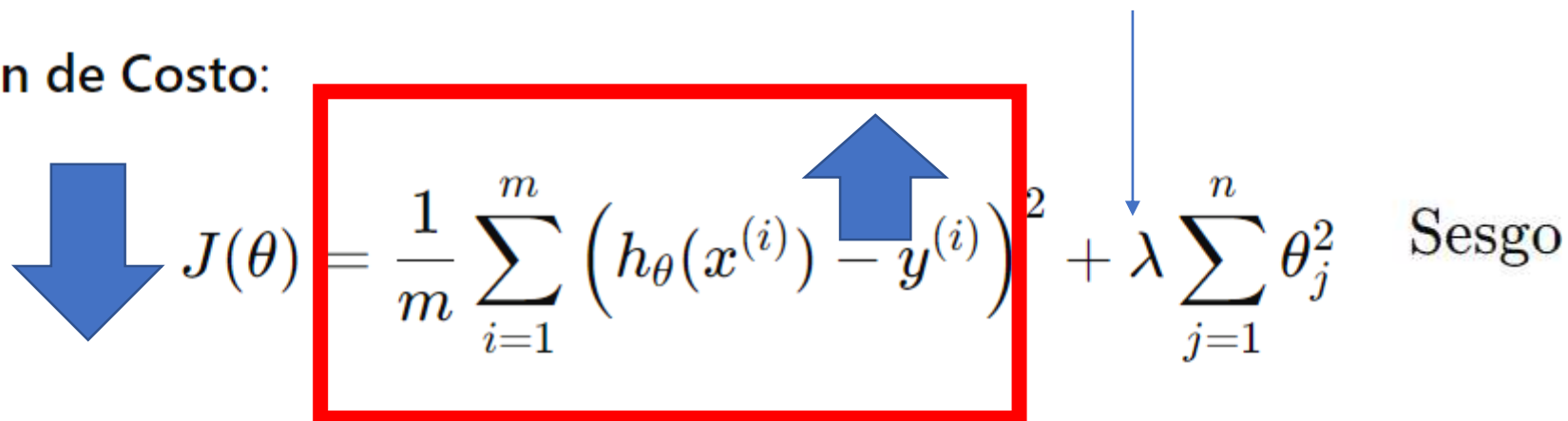


representa el dilema de sesgo-varianza en el aprendizaje automático, mostrando cómo el error total del modelo se descompone en sesgo, varianza y error irreducible, y cómo estos componentes varían con la complejidad del modelo.

Source: <http://scott.fortmann-roe.com/docs/BiasVariance.html>

## Regularización L2 (Ridge)

- **Penalización:** Suma de los cuadrados de los coeficientes ( $\sum \theta_j^2$ ).
- **Efecto:** No elimina completamente los coeficientes, sino que los reduce, lo que puede ser útil para tratar multicolinealidad.
- **Uso:** Se utiliza cuando se espera que todas las características tengan algún grado de relevancia y no se quiere que ningún coeficiente sea exactamente cero.
- **Función de Costo:**



The diagram illustrates the L2 Ridge Cost Function,  $J(\theta)$ . A large blue arrow points down to the function name. The function is defined as the sum of two terms. The first term,  $\frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2$ , is enclosed in a red rectangular box. A blue arrow points up to the minus sign between the hypothesis and the target value. The second term,  $+\lambda \sum_{j=1}^n \theta_j^2$ , has a blue arrow pointing down to the coefficient  $\lambda$ . The word "Sesgo" (Bias) is written to the right of the second term.

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m \left( h_{\theta}(x^{(i)}) - y^{(i)} \right)^2 + \lambda \sum_{j=1}^n \theta_j^2 \quad \text{Sesgo}$$

## Regularización L1 (Lasso)



- **Penalización:** Suma de los valores absolutos de los coeficientes ( $\sum |\theta_j|$ ).
- **Efecto:** Puede llevar a que algunos coeficientes sean exactamente cero, lo que resulta en la selección de características y en modelos más simples.
- **Uso:** Es útil cuando se espera que solo algunas características sean relevantes para el modelo, ya que puede eliminar completamente las características irrelevantes.
- **Función de Costo:**

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m \left( h_{\theta}(x^{(i)}) - y^{(i)} \right)^2 + \lambda \sum_{j=1}^n |\theta_j| \quad \text{Sesgo}$$



## Elastic Net

- **Penalización:** Combina las penalizaciones L1 y L2 ( $\lambda_1 \sum |\theta_j| + \lambda_2 \sum \theta_j^2$ ).
- **Efecto:** Puede tanto eliminar características irrelevantes como reducir los coeficientes de las características relevantes.
- **Uso:** Es útil cuando se sospecha que hay correlación entre las características y se busca un equilibrio entre la selección de características y la reducción de coeficientes.
- **Función de Costo.**

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m \left( h_{\theta}(x^{(i)}) - y^{(i)} \right)^2 + \lambda_1 \sum_{j=1}^n |\theta_j| + \lambda_2 \sum_{j=1}^n \theta_j^2$$

Puede ser una o la otra

Sesgo

## Resumen de Diferencias:

- **Lasso (L1):** Ideal para modelos donde solo algunas características son importantes. Puede reducir los coeficientes de características irrelevantes a cero, eliminándolas efectivamente del modelo.
- **Ridge (L2):** Adecuado para situaciones donde todas las características pueden tener algún grado de relevancia. Reduce la magnitud de todos los coeficientes pero no los lleva a cero.
- **Elastic Net:** Combina las ventajas de Lasso y Ridge, útil en situaciones donde hay muchas características y algunas pueden estar correlacionadas.



```
import numpy as np
from sklearn.linear_model import Ridge, Lasso, ElasticNet
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error

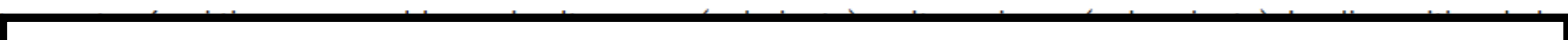
# Generar datos de ejemplo
np.random.seed(42)
X = 2 * np.random.rand(100, 1)
y = 4 + 3 * X + np.random.randn(100, 1)

# Dividir en conjuntos de entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

+ Código + Texto

aplicación dada. En este laboratorio, continuaras con este proceso y explorarás algunos consejos para mejorar el rendimiento de tus modelos.

Resulta que los errores de entrenamiento y validación cruzada pueden indicarte qué hacer a continuación para mejorar tus modelos.

Específicamente,  conferencia

mostrada a contin

La figura de la izq

Como resultado, t

alta varianza, don

rendimiento será

donde el modelo a

algunos consejos

## Resumen:

- **Alto sesgo:** Representado por un modelo simple que no ajusta bien los datos (subajuste, underfitting).
- **Alta varianza:** Representado por un modelo complejo que ajusta demasiado los datos (sobreajuste, overfitting).

Para corregir un problema de alto sesgo, puedes:

- Intentar agregar características polinomiales
- Intentar obtener características adicionales
- Intentar disminuir el parámetro de regularización
- Para corregir un problema de alta varianza, puedes:
- Intentar aumentar el parámetro de regularización
- Intentar conjuntos de características más pequeños
- Obtener más ejemplos de entrenamiento
- Probarás todos estos consejos en este laboratorio. ¡Empecemos!

avianca - Avianca ofrecerá **más de 34.000 vuelos** y espera transportar alrededor de 4.7 millones de clientes durante la temporada alta de mitad de año.

Regularización : parámetros que escalan las características multiplican por un valor 1 o cero  
Tratar de tener modelos mas simples ( disminuir la amplitud para dejar que el modelo se ajuste mas a los datos)

Modelo de alta varianza no generaliza , muy complejo aumentarle la regularización con los coeficientes

Eliminar el ruido

Puedo tener los dos problemas en un sobreajuste puede ser que tenga sobre ajuste porque tengo una ecuación muy compleja en la regresión, o sea un polinomio muy grande o puede ser que es que no es que el polinomio sea tan grande, sino que me faltan datos de entrenamiento.

```

model = LinearRegression()

# Train and plot polynomial regression models
train_plot_poly(model, x_train, y_train, x_cv, y_cv, max_degree=10, baseline=400)

```



SHAPE 60 OBSERVACIONES CON UNA CARACTERISTICA

the shape of the training set (input) is: (60, 1)  
the shape of the training set (target) is: (60,)

the shape of the cross validation set (input) is: (20, 1)  
the shape of the cross validation set (target) is: (20,)

Validación cruzada, si tengo 1000 datos, entonces yo primero uso los primeros 100 para validar y de los otros 900 para entrenar y luego cojo los siguientes 100 de 100 a 200.)-MINIMIZA EL SESGO

\*Cada que itera

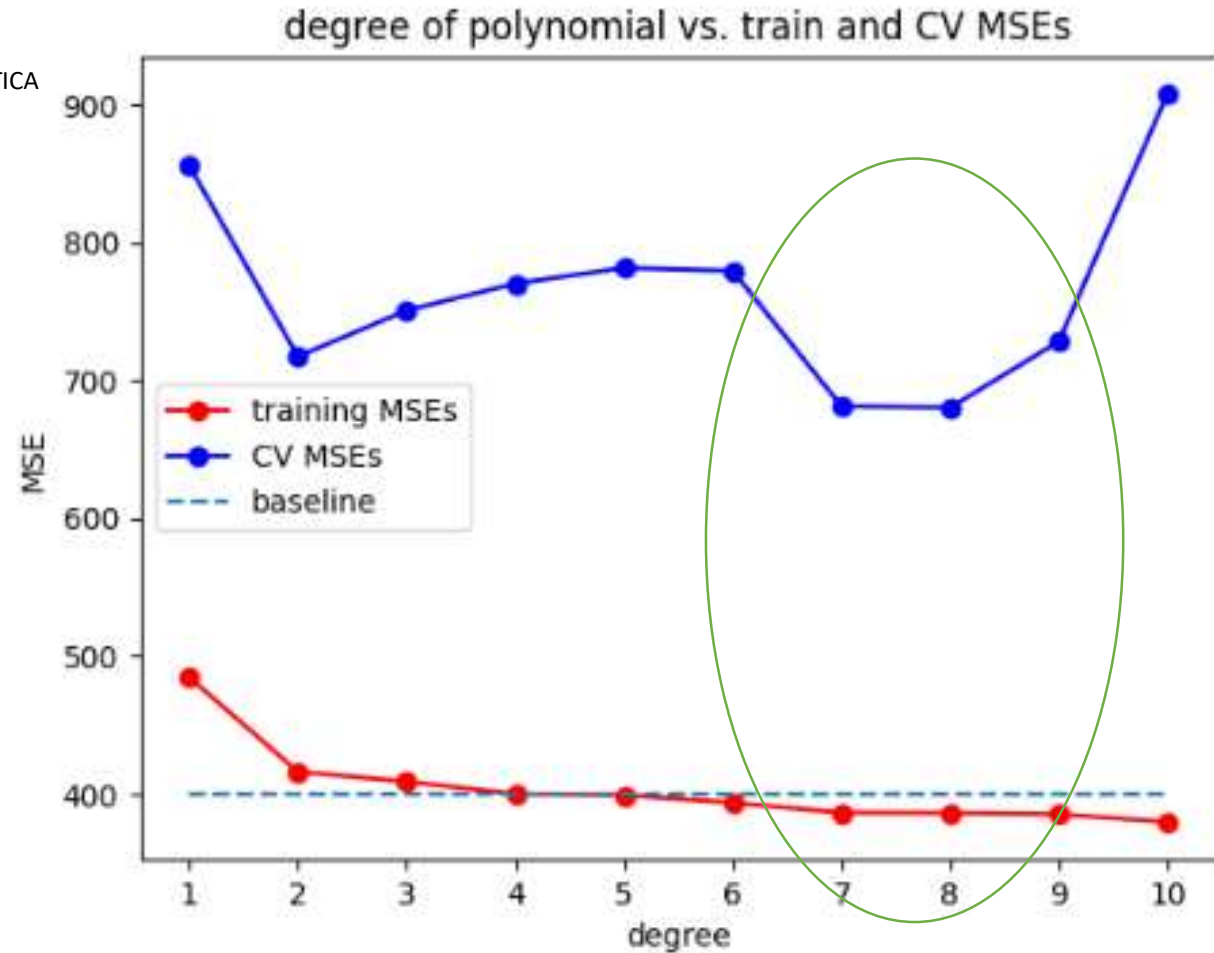
```

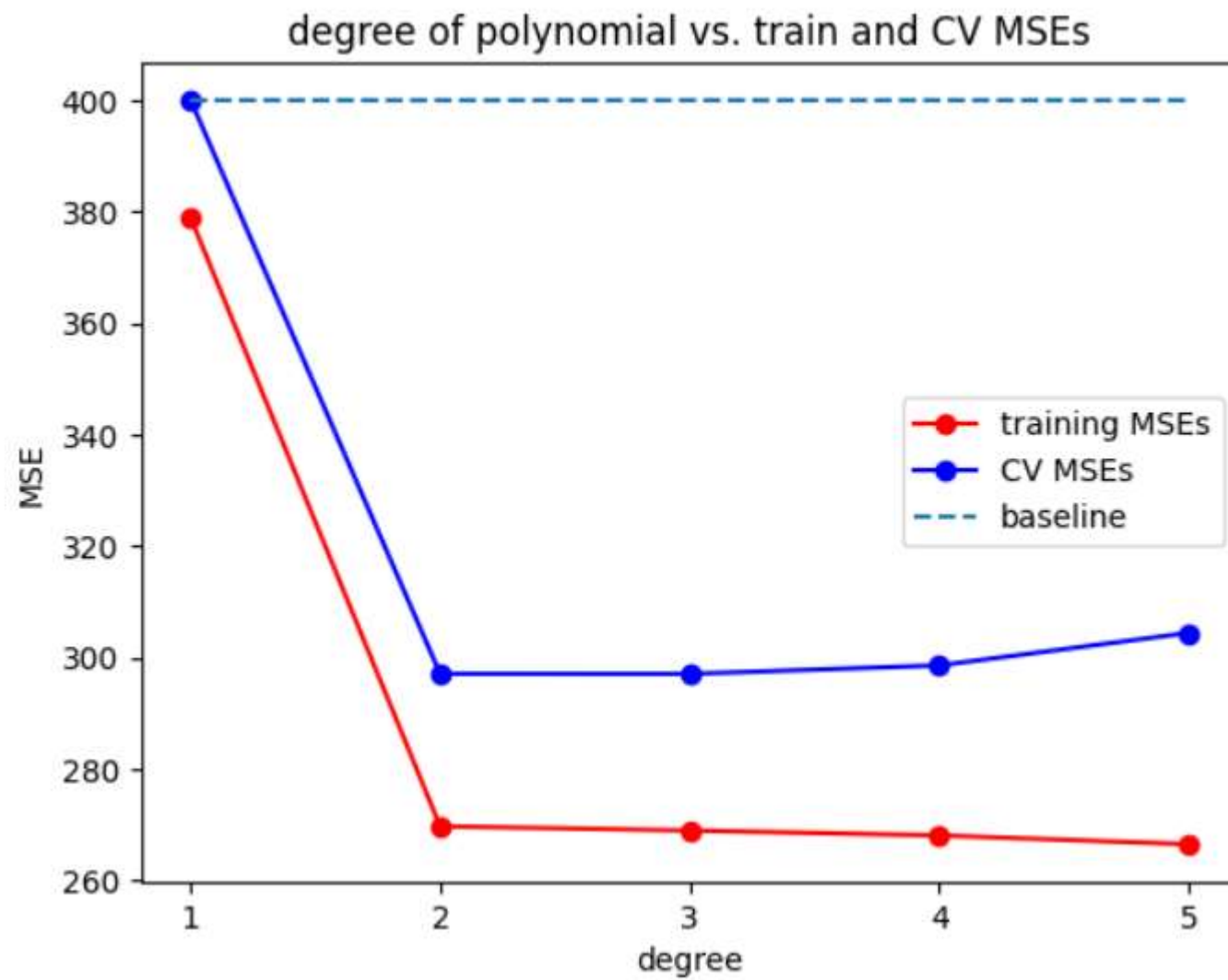
def cross_validation(model, X, y, k=5):
    """
    Realiza la validación cruzada sobre un modelo dado.

    Parámetros:
    - model: el modelo de aprendizaje automático a validar.
    - X: características de entrada (matriz de datos).
    - y: etiquetas o valores objetivo.
    - k: número de folds (por defecto 5).

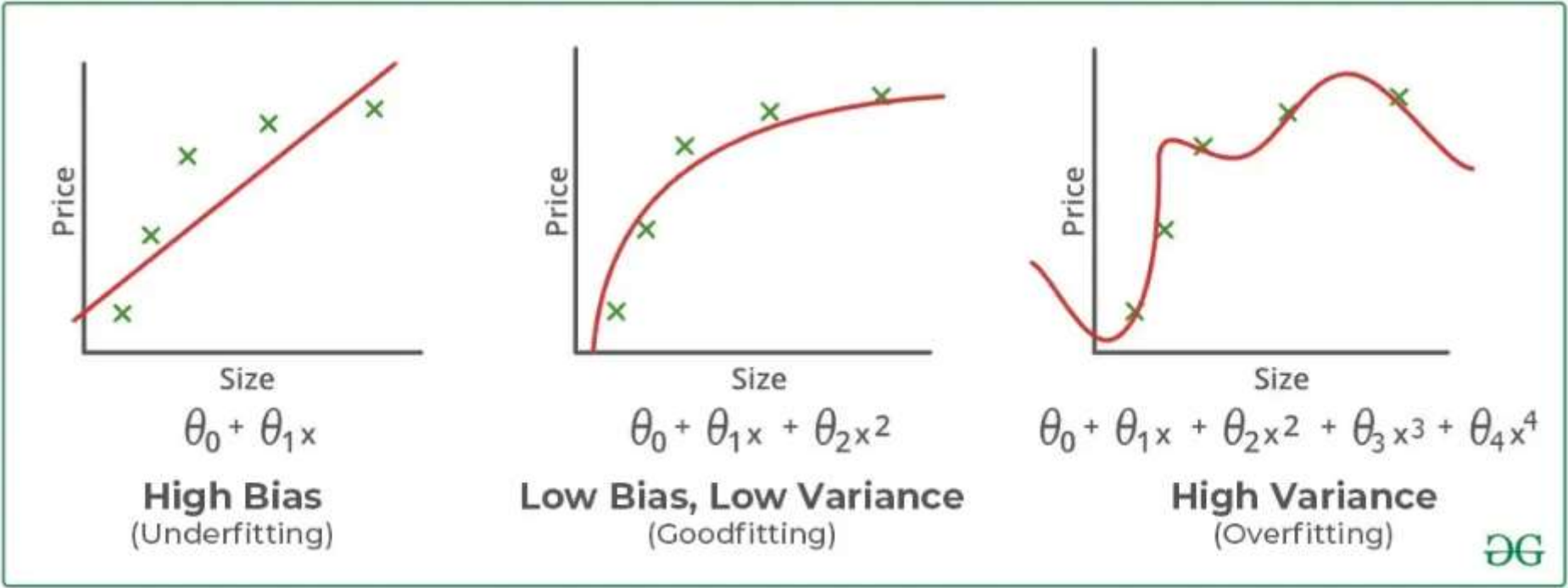
    Retorna:
    - El error promedio de validación cruzada (MSE).
    """

```





MORE FEATURES TO TREAT HIGH BIAS:



2 features  
e=6, baseline=250)

the shape of  
the shape of

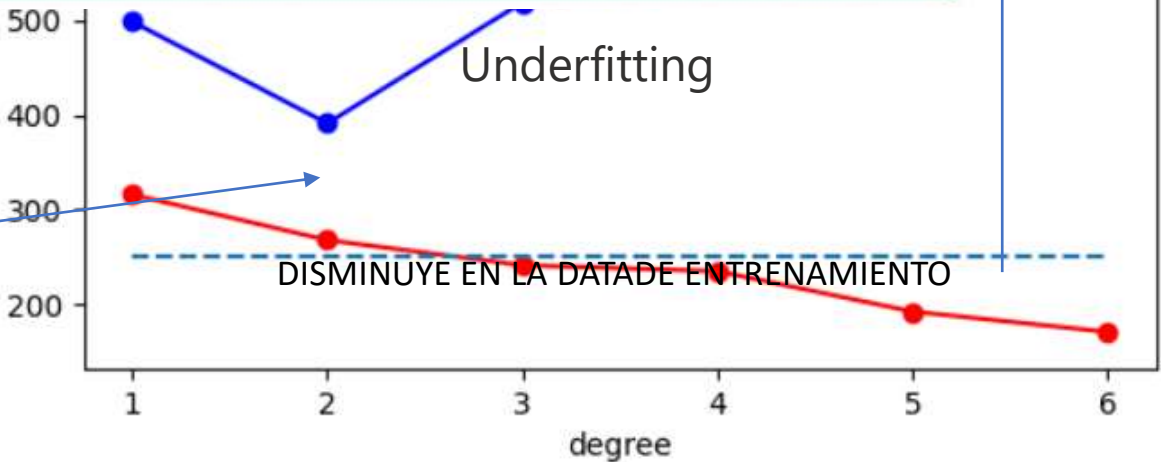
the shape of  
the shape of

first 5 rows  
[[3.75757576  
[2.87878788e  
[3.54545455e+03 3.71717172e+00]  
[1.57575758e+03 5.97979798e+00]  
[1.66666667e+03 1.61616162e+00]]

Error total bajo

¿ Cual es el polinomio ideal ?

Esto es el modelo ideal porque tiene un error bajo en el  
entrenamiento y un error bajo en la validación





```

#Define the model
model = LinearRegression()

#Define properties of the 2 datasets
file1 = {'filename':f"{path}{filename3}", 'label': '3 features', 'linestyle': 'd'}
file2 = {'filename':f"{path}{filename2}", 'label': '2 features', 'linestyle': 's'}
files = [file1, file2]

#Train and plot for each dataset
train_plot_diff_datasets(model, files, max_degree=4, baseline=250)

```

