# Análisis de Datos con Python

## Importación de Datos

### *Librerías principales*

- Computación Científica: Pandas, Numpy, Scipy
- Visualización: Matplotlib, Seaborn
- Modelos: Scikit-learn, Statsmodels

### *Importar data*

- Debemos tener en cuenta dos conceptos importantes:
  - Formato: .csv | .json, .xlsx, ….
  - Ruta de los Datos: pueden estar almacenados localmente o en internet.

### *Conocimientos Básicos sobre los datos*

- Antes de analizar, es fundamental entender la estructura de la data
  - Revisar el tipo de datos y su distribución.
  - Detectar posibles problemas o inconsistencias.
- ¿Por qué revisar el tipo de ero puede cometer errores. C as a cada columna.

| Pandas Type | Native Python Type | Description |
|---|---|---|
| object | string | numbers and strings |
| int64 | int | Numeric characters |
| float64 | float | Numeric characters with decimals |
| datetime64, timedelta[ns] | N/A (but see the datetime module in Python's standard library) | time data. |

## Accediendo a la base de datos

Python se conecta a los sistemas gestores de bases de datos (DBMS) mediante llamadas a APIs.

### *¿Qué es un API SQL?*

Una API SQL permite a una aplicación enviar sentencias SQL al DBMS y recuperar resultados. Pasos generales:

1. Conectarse al DBMS mediante la API.
2. Crear y enviar sentencias SQL.
3. Consultar el estado y gestionar errores.
4. Cerrar la conexión.

### *API de bases de datos en Python*

Es la interfaz estándar para acceder a bases de datos relacionales. Permite que el mismo código funcione con diferentes DBMS.

a. Objetos de conexión: manejan la conexión y las transacciones.
b. Objetos de cursor: ejecutan consultas


### *Métodos de Conexión*

- Método del cursor - cursor(): Devuelve un nuevo objeto de cursor mediante la conexión.
- Método de confirmación - commit(): Se utiliza para confirmar cualquier transacción.
- Método de reversión - rollback(): La base de datos revierte transacciones pendientes.
- Método cerrado - close(): Se utiliza para cerrar una conexión a la base de datos.

## Manipulación de Datos

También llamada "limpieza" o "preprocesamiento". Consiste en transformar los datos crudos en un formato adecuado para el análisis.

### *Objetivos*

| Manejar valores faltantes | Formatear los datos | Normalizar valores |

| Agrupar datos (binning) | Convertir variables categóricas a numéricas |

### *Valores perdidos*

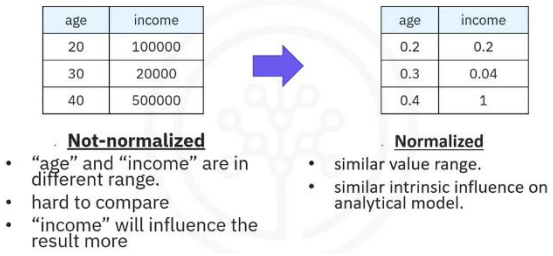
- Pueden representarse como "?", "N/A", 0 o celdas vacías. Opciones para tratarlos
  - Consultar la fuente para recuperar el dato.
  - Eliminar la fila o columna.
  - Reemplazar por media, moda o estimaciones.
  - Dejarlos como están.

### *Formateo de Datos*

- Los datos son usualmente recogidos de diferentes lugares y guardados en diferentes formas, el objetivo nuestro es manipular los datos para convertirlos en expresiones simples y comunes para hacer una comparación significativa.


### *Normalización de Datos*

- Ayuda a que todas las variables tengan un rango comparable.
  Técnicas:
  - Escalado simple: $x_{new} = \frac{x_{old}}{x_{max}}$.
  - Min-Max: $x_{new} = \frac{x_{old} - x_{min}}{x_{max} - x_{min}}$
  - Z-score: $x_{new} = \frac{x_{old} - \mu}{\sigma}$




### *Agrupamiento [Binning]*

- Consiste en agrupar valores numéricos en categorías. Ejemplo: Precios de 0 a 100 soles [barato], de 100 a 250 [medio], de 250 a más [caro]


### *Variables Categóricas a Variables Cuantitativas*

- Los modelos no pueden usar strings como entrada. Se crean variables dummy para representar cada categoría.

| Car | Fuel | ... | gas | diesel |
|---|---|---|---|---|
| A | gas | ... | 1 | 0 |
| B | diesel | ... | 0 | 1 |
| C | gas | ... | 1 | 0 |
| D | gas | ... | 1 | 0 |

## Análisis Exploratorio de Datos

Sirve para resumir y entender los datos. Se busca descubrir relaciones y variables importantes.

### *Estadísticas Descriptivas*

- Permiten comprender las características básicas de los datos.
    - Diagramas de caja: resumen de variables numéricas.
    - Gráficos de dispersión: muestran la relación entre dos variables (x: independiente, y: dependiente).

### *Agrupación de Datos*

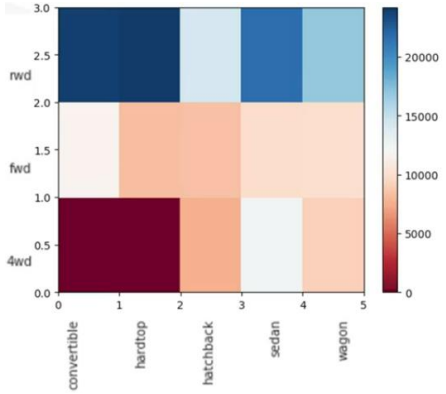
- Se analizan las relaciones entre una variable categórica y una numérica. Ejemplo: tipo de vehículo vs. precio.

| | drive-wheels | body-style | price |
|---|---|---|---|
| 0 | 4wd | hatchback | 7603.000000 |
| 1 | 4wd | sedan | 12647.333333 |
| 2 | 4wd | wagon | 9095.750000 |
| 3 | fwd | convertible | 11595.000000 |
| 4 | fwd | hardtop | 8249.000000 |
| 5 | fwd | hatchback | 8396.387755 |
| 6 | fwd | sedan | 9811.800000 |
| 7 | fwd | wagon | 9997.333333 |
| 8 | rwd | convertible | 23949.600000 |
| 9 | rwd | hardtop | 24202.714286 |
| 10 | rwd | hatchback | 14337.777778 |
| 11 | rwd | sedan | 21711.833333 |
| 12 | rwd | wagon | 16994.222222 |

GroupBy Table

| | price | | | | |
|---|---|---|---|---|---|
| body-style | convertible | hardtop | hatchback | sedan | wagon |
| drive-wheels | | | | | |
| 4wd | 0.0 | 0.000000 | 7603.000000 | 12647.333333 | 9095.750000 |
| fwd | 11595.000000 | 8429.000000 | 8396.387755 | 9811.8000000 | 9997.333333 |
| rwd | 23949.600000 | 24202.714286 | 14337.777778 | 21711.833333 | 16994.222222 |

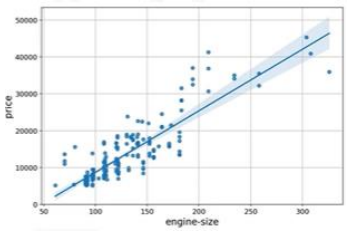
Pivot Table




Heatmap

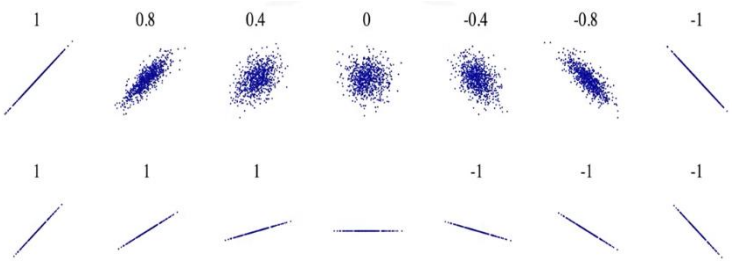
### *Correlación de Datos*

- Mide la dependencia entre variables.
    - Correlación lineal negativa.
    - Correlación lineal positiva.
    - No correlación.




### *Correlación de Pearson*

- Mide la fuerza de la correlación entre dos variables.
    - Coeficiente de Correlación:
        - $CC = 1$, correlación perfecta positiva
        - $CC = -1$, correlación perfecta negativa
        - $CC = 0$, no correlación
    - P-Value:
        - P-Value $< 0.001$, certeza fuerte de correlación.
        - P-Value < 0.05, certeza moderada de correlación.
        - P-Value < 0.1, certeza débil de correlación.



## Desarrollo de Modelos

- Un modelo o estimador es una ecuación matemática que se utiliza para predecir un valor dado uno o varios valores distintos relacionando una o más variables o características independientes con variables dependientes.

Tamaño del Carro → Modelo → Predecir precio

- Usando la data más relevante posible tendremos un modelo más preciso.

### *Modelo de Regresión Lineal Simple*

- Modelo con una variable independiente. Forma:

$$y = b_0 + b_1 x_1$$

$b_0$ es el intercepto, $b_1$ es nuestro parámetro de interés (pendiente).

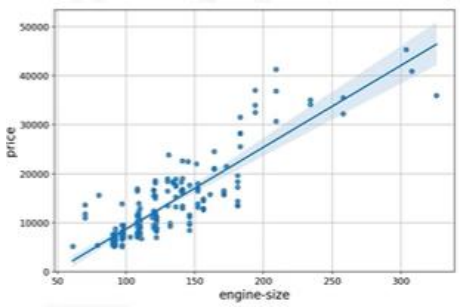
### *Modelo de Regresión Lineal Múltiplo*

- Modelo con varias variables independientes. Forma:

$$y = b_0 + b_1 x_1 + b_2 x_2 + b_3 x_3 + \cdots + b_k x_k$$

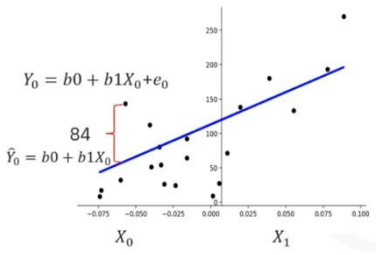
### *Gráficos de Regresión*

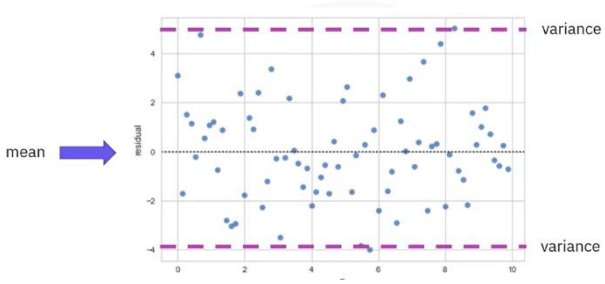
- Es una combinación entre el gráfico de dispersión y la línea de regresión ajustada.
- Son una buena forma de estimar:
  - La relación entre dos variables.
  - La fuerza de correlación.
  - La dirección (positiva o negativa).




### *Gráficos de Residuos*

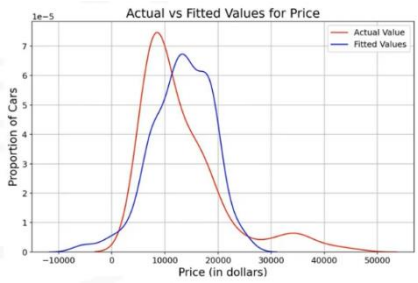
- Muestra la diferencia entre valores reales y predichos. Idealmente, los errores deben distribuirse aleatoriamente.
- Obtenemos el residuo restando el valor previsto y el valor objetivo real.



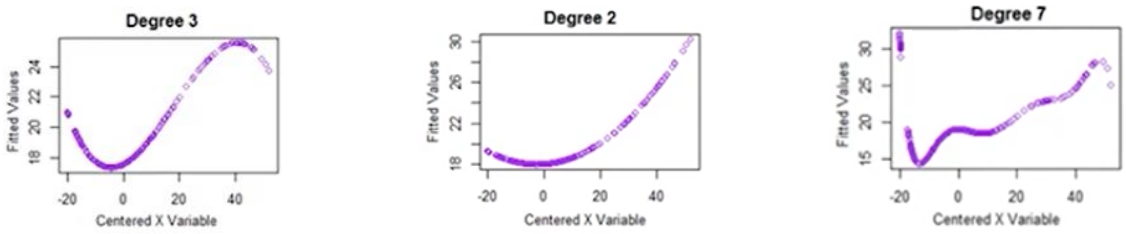

### *Gráfico de Distribución*

- Muestra el valor previsto en comparación con el valor verdadero.
  - La línea roja es el valor real.
  - La línea azul es el valor estimado.
- Es una buena comparación para saber la precisión de nuestro modelo.



### *Regresión Polinomial*

- Se usa cuando la relación no es lineal, sino curvilínea. La relación curvilínea se obtiene elevando al cuadrado o fijando términos de orden superior a las variables predictoras. Ejemplos:
  - Cuadrático (2do orden): $y = b_0 + b_1x_1 + b_2(x_1)^2$
  - Cúbico (3er orden): $y = b_0 + b_1x_1 + b_2(x_1)^2 + b_3(x_1)^3$
  - Orden mayor: $y = b_0 + b_1x_1 + b_2(x_1)^2 + b_3(x_1)^3 + \cdots$



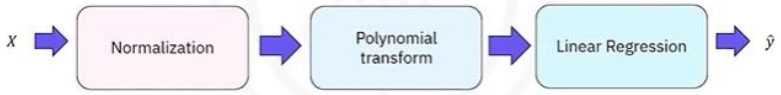

### *Regresión polinomial con más de una dimensión*

- Tiene la siguiente forma:

$$y = b_0 + b_1x_1 + b_2(x_2)^2 + b_3(x_3)^3 + +b_4x_1x_2 \ldots$$

### *Pipelines*

- Permiten encadenar transformaciones y modelado en un solo flujo:

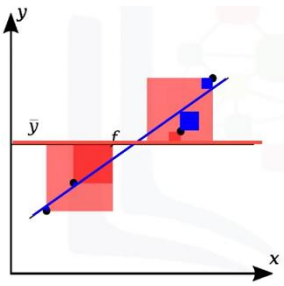



### *Medidas para Evaluar la Muestra*

- Para saber qué también se ajusta el modelo al conjunto de datos usamos dos medidas:
  - Mean Squared Error (MSE): Promedio de todas las diferencias.
  - R-squared ($R^2$): Mide qué tan cerca la data se ajusta a la regresión lineal. Es el pocentaje de variación de la variable dependiente explicada por la data.
    - Cálculos:

$$R^2 = \left(1 - \frac{MSE\ of\ Regresion\ Line}{MSE\ of\ the\ average\ of\ the\ data}\right)$$

$$\frac{MSE\ of\ regresion\ line}{MSE\ of\ \acute{y}} = \frac{\sum azul}{\sum rojo}$$




- The blue line represents the regression line
- The blue squares represents the MSE of the regression line
- The red line represents the average value of the data points
- The red squares represent the MSE of the red line
- We see the area of the blue squares is much smaller than the area of the red squares

## Evaluación del Modelo y Refinamiento

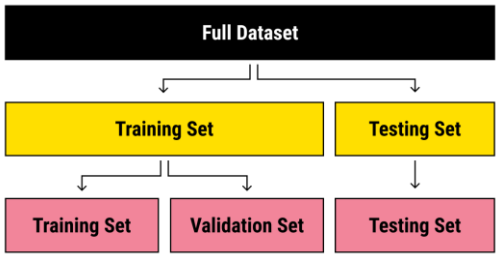
La evaluación del modelo nos indica cómo funciona nuestro modelo en el mundo real.

### *Evaluación en la Muestra*

- La evaluación por muestreo nos indica en qué medida nuestro modelo se ajusta a los datos ya proporcionados para entrenarlo.
    - Problema: No nos proporciona una estimación de qué tan bien el modelo entrenado puede predecir nuevos datos.
    - Solución: Consiste en dividir los datos, utilizar los datos de la muestra para entrenar el modelo. El resto de los datos, denominados datos de prueba, se utilizan como datos fuera de la muestra. Estos últimos se utilizan luego para aproximar el rendimiento del modelo en el mundo real.

### *Training/Testing Sets*

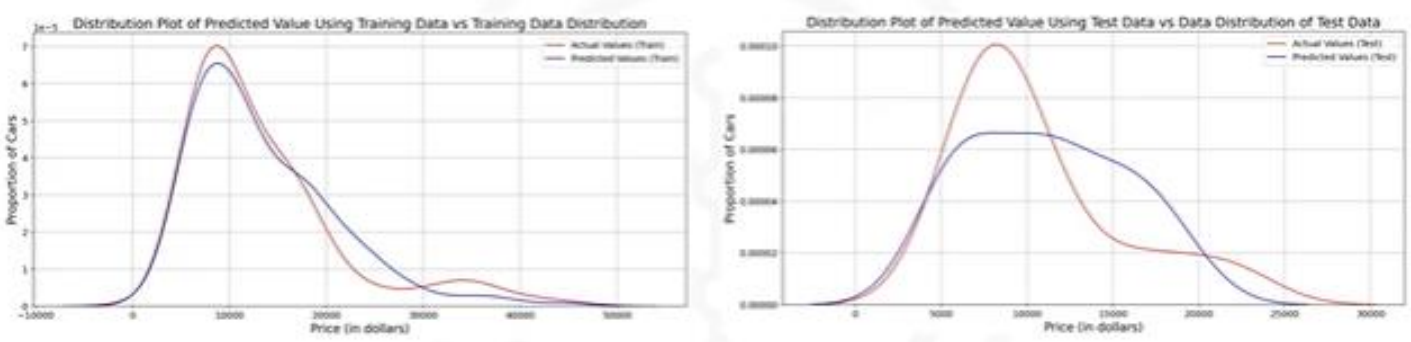
- Es el proceso de separar los datos en conjuntos de entrenamiento y pruebas es una parte importante de la evaluación del modelo.
    - Cuando dividimos los datos, normalmente la mayor parte de los datos se utiliza para el entrenamiento y una parte más pequeña para las pruebas.




Usamos un conjunto de entrenamiento para construir un modelo y descubrir relaciones predictivas. Luego utilizamos un conjunto de pruebas para evaluar el rendimiento del modelo. Cuando hayamos terminado de probar nuestro modelo, debemos usar todos los datos para entrenarlo.

### *Error generalizado*

- El error de generalización es una medida de la eficacia con la que nuestros datos predicen datos nunca antes vistos. El error que obtenemos con los datos de nuestras pruebas es una aproximación de este error.




La figura de la izquierda muestra la distribución de los valores reales de la data de entrenamiento en rojo en comparación con los valores pronosticados a partir de una regresión lineal en azul. Vemos que las distribuciones son algo similares. Si generamos la misma gráfica con los datos de prueba, vemos que las distribuciones son relativamente diferentes. La diferencia se debe a un ***error de generalización*** y representa lo que vemos en el mundo real.

*Pérdida de datos de entrenamiento*

Cuando usamos una gran proporción de datos para entrenar el modelo (por ejemplo, 90%) y dejamos una parte pequeña para pruebas (10%), obtenemos una estimación razonable del rendimiento. Sin embargo, esta estimación tiene poca precisión: si repetimos el experimento con diferentes combinaciones de entrenamiento y prueba, los resultados varían considerablemente.

Por ejemplo, supongamos que repetimos el proceso varias veces con diferentes divisiones aleatorias de los datos. Cada vez, obtenemos un resultado distinto. En conjunto, estas estimaciones se acercan al verdadero error de generalización, pero su dispersión es alta: no son consistentes entre sí.

Por otro lado, si usamos menos datos para entrenar y más para probar, los resultados de evaluación serán más consistentes entre sí (mayor precisión), pero el modelo en sí será menos preciso, ya que se entrenó con menos información.

En resumen:

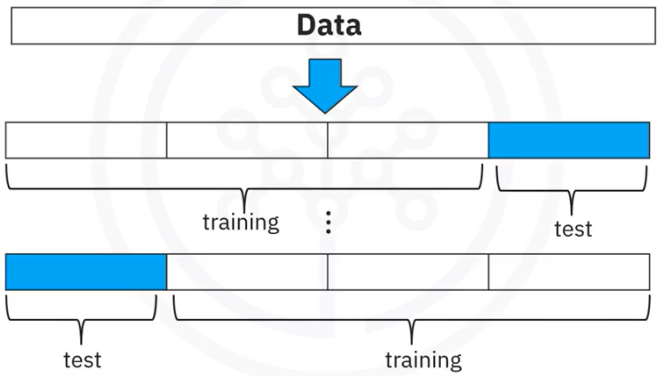
- Más datos para entrenar → modelo más preciso, pero evaluación menos confiable.
- Más datos para probar → evaluación más precisa, pero modelo menos preciso.

*¿Cómo se soluciona esto?*

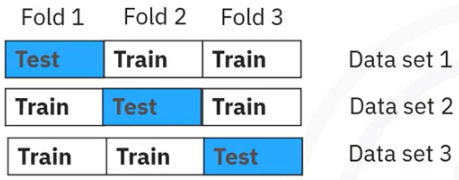
Usamos validación cruzada. Este método permite usar todos los datos tanto para entrenamiento como para prueba, en diferentes combinaciones, lo que mejora tanto la precisión del modelo como la confiabilidad de la evaluación.

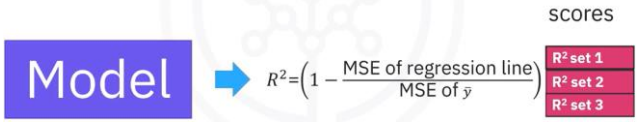
*Validación Cruzada*

- En este método, el conjunto de datos se divide en k grupos iguales. Cada grupo se denomina pliegue, por ejemplo, cuatro pliegues.
- Algunos de los pliegues se pueden usar como un conjunto de entrenamiento que usamos para entrenar el modelo, y las partes restantes se usan como un conjunto de prueba que usamos para probar el modelo.




- Ejemplo:
  - Podemos usar tres pliegues para el entrenamiento y luego usar un pliegue para las pruebas. Esto se repite hasta que cada partición se utilice tanto para el entrenamiento como para las pruebas.
  - Al final, utilizamos los resultados promedio como estimación del error fuera de la muestra. La métrica de evaluación depende del modelo. Por ejemplo, el $R^2$.

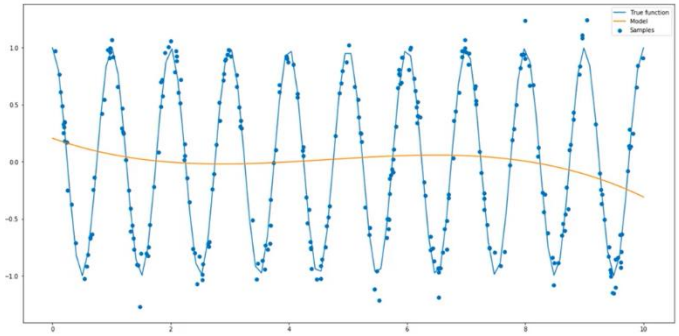





- cross_val_score: devuelve el score promedio.
- cross_val_predict: devuelve los valores predichos para comparar con los reales


*Elección de Modelo*

**Definición del Problema**: Se busca determinar el mejor orden de un polinomio para estimar una función con datos ruidosos. Un modelo demasiado simple genera errores altos (subajuste), mientras que uno demasiado complejo sigue el ruido en lugar de la tendencia real (sobreajuste).



- **Subajuste**: Ocurre cuando el modelo es demasiado simple, como al usar una regresión lineal para datos que requieren mayor complejidad. Esto resulta en un mal ajuste y altos errores.
- **Sobreajuste**: Un polinomio de orden muy alto se ajusta excesivamente a los datos de entrenamiento, capturando ruido en lugar de la tendencia real. Esto deteriora el desempeño en datos nuevos.
- **Selección del Mejor Modelo**: Se analiza el error cuadrático medio (ECM) en datos de entrenamiento y prueba. El ECM de prueba se minimiza en un orden óptimo del polinomio. En el ejemplo dado, el mejor ajuste se obtuvo con un polinomio de grado 8.
- **Errores en la Modelación**: Existen errores irreducibles debido al ruido en los datos y errores derivados de una mala especificación del modelo (e.g., usar un polinomio para datos sinusoidales).
- **Evaluación con $R^2$**: Para validar la elección del modelo, se usa $R^2$. En el ejemplo de predicción de precios según caballos de fuerza, el mejor ajuste se obtuvo con un polinomio de grado 3, ya que $R^2$ disminuyó al usar un polinomio de grado 4.

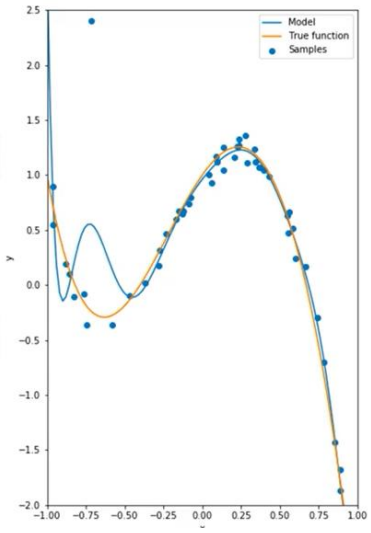
*Ridge Regression (Regresión de Crestas)*

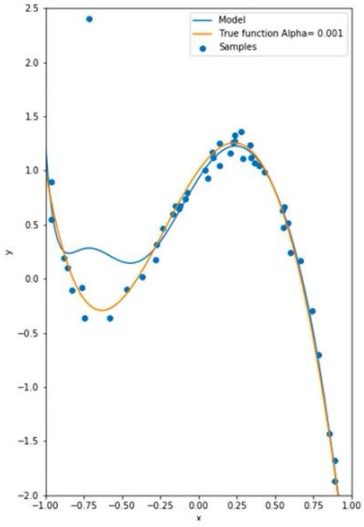
Es una técnica de regresión lineal regularizada que añade un término de penalización (L2) a la función de pérdida para controlar el sobreajuste causado por multicolinealidad o uso de características polinomiales de alto orden. Motivación:

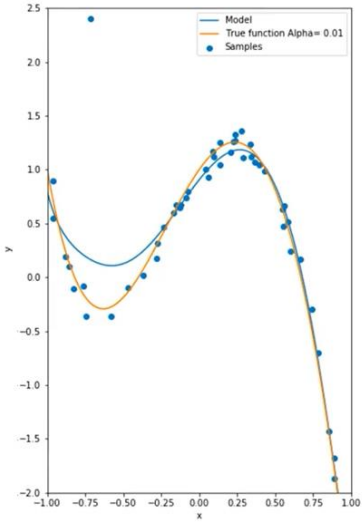
- Modelos con muchas variables o polinomios de alto grado tienden a sobreajustar.
- Esto ocurre especialmente con multicolinealidad (variables altamente correlacionadas).
- El sobreajuste genera coeficientes grandes, lo que aumenta el error estándar y reduce la generalización del modelo.
- Ridge controla esto reduciendo la magnitud de los coeficientes

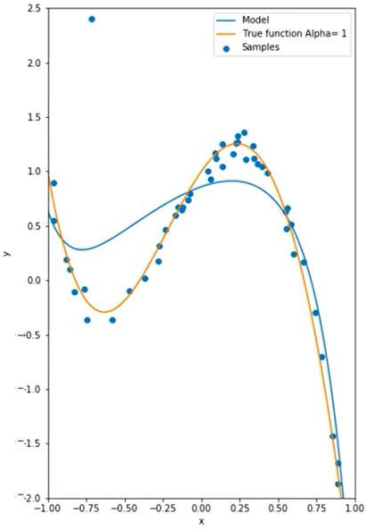
$$\hat{y} = 1 + 2\,x - 3\,x^2 - 2x^3 - 12\,x^4 - 40x^5 + 80x^6 + 71x^7 - 141x^8 - 38x^9 + 75x^{10}$$

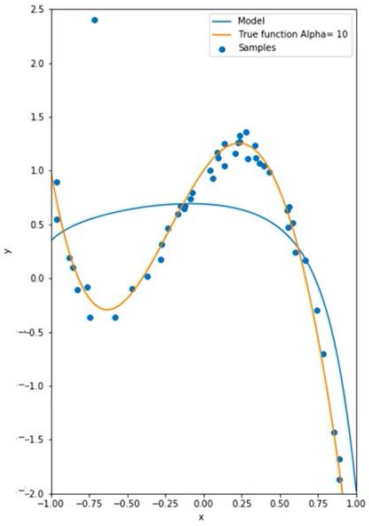
| ***Alpha*** | $x$ | $x^2$ | $x^3$ | $x^4$ | $x^5$ | $x^6$ | $x^7$ | $x^8$ | $x^9$ | $x^{10}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2 | -3 | -2 | -12 | -40 | 80 | 71 | -141 | -38 | 75 |
| 0.001 | 2 | -3 | -7 | 5 | 4 | -6 | 4 | -4 | 4 | 6 |
| 0.01 | 1 | -2 | -5 | -0.04 | 0.15 | -1 | 1 | -0.5 | 0.3 | 1 |
| 1 | 0.5 | -1 | -1 | -0.614 | 0.70 | -0.38 | -0. 56 | -0.21 | -0.5 | -0.1 |
| 10 | 0 | -0.5 | -0.3 | -0.37 | -0.30 | -0.30 | -0.22 | -0.22 | -0.22 | -0.17 |



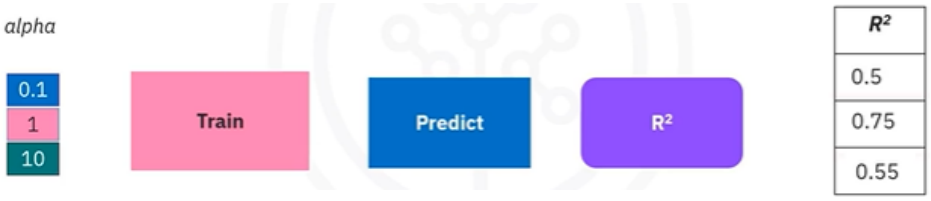




Podemos observar que a medida que 0.1 es la mejor penalización que ese le puede poner al modelo.

Procedimiento:

1. Seleccionar conjunto de datos
    a. Entrenamiento
    b. Validación (NO confundir con test)
2. Inicializar valores de α (por ejemplo: 0, 0.001, 0.01, 1, 10...)
3. Para cada valor de α:
    a. Entrenar modelo Ridge con Ridge(alpha=...)
    b. Predecir sobre el conjunto de validación
    c. Calcular métrica (ej. $R^2$ o MSE)
    d. Guardar resultados
4. Seleccionar el mejor α:
    a. Aquel que maximiza $R^2$ (o minimiza MSE) en datos de validación.




Ideas clave:

- Un α pequeño no corrige el sobreajuste.
- Un α grande produce subajuste (coeficientes ≈ 0).
- Ridge es ideal para datos con muchas características correlacionadas o modelos polinomiales.
- Siempre usar validación cruzada para seleccionar el mejor α.
- Puede usarse con otras métricas: MSE, MAE, etc.
- No elimina variables (como Lasso), solo reduce sus coeficientes.
- Utiliza mejor generalización a costa de perder algo de ajuste en entrenamiento

## *Grid Search*

- Automatiza la búsqueda del mejor hiperparámetro (como $\alpha$).
- Prueba todas las combinaciones posibles en una cuadrícula (grid).
- Usa validación cruzada para evaluar cada combinación.

```
from sklearn.linear_model import Ridge
from sklearn.model_selection import GridSearchCV

parameters1= [{'alpha': [0.001,0.1,1, 10, 100, 1000,10000,100000,100000]}]

RR=Ridge()

Grid1 = GridSearchCV(RR, parameters1,cv=4)

Grid1.fit(x_data[['horsepower', 'curb-weight', 'engine-size',
'highway-mpg']],y_data)

Grid1.best_estimator_

scores  = Grid1.cv_results_
scores['mean_test_score']
```

```
for param,mean_val, mean_test in zip(scores['params'], scores['mean_test_score'],
scores['mean_train_score']):

print(param, "R^2 on test data:", mean_val,"R^2 on train data:", mean_test)
```

```
{'alpha': 0.001, 'normalize': True} R^2 on tesst data: 0.66605547293 R^2 on train data: 0.814001968709
{'alpha': 0.001, 'normalize': False} R^2 on tesst data: 0.665488366584 R^2 on train data: 0.814002698797
{'alpha': 0.1, 'normalize': True} R^2 on tesst data: 0.694175625356 R^2 on train data: 0.810546768311
{'alpha': 0.1, 'normalize': False} R^2 on tesst data: 0.665488937796 R^2 on train data: 0.814002698794
{'alpha': 1, 'normalize': True} R^2 on tesst data: 0.690486934584 R^2 on train data: 0.749104440368
{'alpha': 1, 'normalize': False} R^2 on tesst data: 0.665494127178 R^2 on train data: 0.814002698472
{'alpha': 10, 'normalize': True} R^2 on tesst data: 0.321376875232 R^2 on train data: 0.341856042902
{'alpha': 10, 'normalize': False} R^2 on tesst data: 0.665545680812 R^2 on train data: 0.8140026666
{'alpha': 100, 'normalize': True} R^2 on tesst data: 0.0170551710263 R^2 on train data: 0.0496044796826
{'alpha': 100, 'normalize': False} R^2 on tesst data: 0.666029359996 R^2 on train data: 0.813999791851
{'alpha': 1000, 'normalize': True} R^2 on tesst data: -0.0301961745066 R^2 on train data: 0.005184451599
{'alpha': 1000, 'normalize': False} R^2 on tesst data: 0.668968215369 R^2 on train data: 0.813870488264
{'alpha': 10000, 'normalize': True} R^2 on tesst data: -0.0351687400461 R^2 on train data: 0.000520784757979
{'alpha': 10000, 'normalize': False} R^2 on tesst data: 0.673346359342 R^2 on train data: 0.812583743226
{'alpha': 100000, 'normalize': True} R^2 on tesst data: -0.0356685844558 R^2 on train data: 5.2101975528e-05
```