Ingeniería de Características

Víctor Aceña - Isaac Martín

DSLAB

2025-09-11

La Ingeniería de Características como Arte y Ciencia

La **ingeniería de características** es el proceso fundamental que transforma y crea variables para maximizar la capacidad predictiva y la interpretabilidad de los modelos.

El problema:

- Los datos raramente están en forma óptima
- Relaciones no lineales ocultas
- Variables categóricas sin procesar
- Efectos de interacción ignorados

La solución:

- Transformaciones matemáticas precisas
- Codificación inteligente de categorías
- Creación de interacciones significativas
- Combinaciones y ratios informativos

Principio clave: "Los datos y la preparación de características determinan el límite superior del rendimiento; los modelos solo se aproximan a ese límite" - *Andrew Ng*

Objetivos de Aprendizaje

1. **Identificar cuándo aplicar transformaciones** específicas según el problema detectado
2. **Aplicar transformaciones clásicas y avanzadas** (logarítmica, Box-Cox, Yeo-Johnson) apropiadamente
3. **Interpretar modelos transformados** comprendiendo cómo cambian los coeficientes
4. **Codificar variables categóricas** usando ordinal encoding y one-hot encoding según su naturaleza
5. **Crear e interpretar interacciones** entre variables continuas, categóricas y mixtas
6. **Aplicar ingeniería avanzada** mediante combinaciones, ratios y transformaciones compuestas

Diagnóstico: ¿Cuándo Transformar?

- **Principio Clave:** Diagnosticar antes de transformar.
- **Riesgos de una Mala Práctica:**
  - **Sobreajuste:** Se pierde capacidad de generalización.
  - **Pérdida de Interpretabilidad:** Se aplican transformaciones sin base teórica.
  - **Violación de Supuestos:** Solucionar un problema creando otro.
  - **Sesgo de Selección:** Elegir por "mejor resultado" sin justificación.

Regla de oro: No transformes sin un diagnóstico previo. Cada cambio debe estar justificado.

El Enfoque Metodológico

- **Clave:** Proceso sistemático basado en evidencia, no solo en métricas de ajuste (como el R^2).
- **Principios de Actuación:**
  1. **Diagnóstico Previo:** Análisis visual y estadístico.
  2. **Justificación Teórica:** Base conceptual para cada transformación.
  3. **Evaluación Integral:** Medir ajuste, interpretabilidad y robustez.
  4. **Validación Posterior:** Verificar la solución sin crear nuevos problemas.
  5. **Parsimonia:** Preferir siempre la solución más simple.

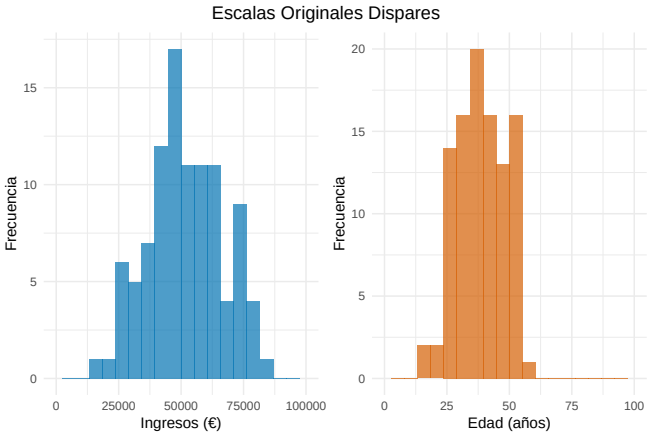
Escalado y Normalización

¿Por qué escalar?

- **Comparabilidad:** Coeficientes en misma escala
- **Regularización:** Penalización justa en Ridge/Lasso
- **Convergencia:** Optimización más eficiente
- **Interpretación:** Efectos estandarizados

Problema común:

Variables con escalas muy diferentes pueden dominar el modelo y hacer que los coeficientes no sean comparables entre sí.



Estandarización (Z-Score)

$$X_{std} = \frac{X - \bar{X}}{\sigma_X}$$

Es la técnica más común. Transforma los datos para que tengan una **media de 0** y una **desviación estándar de 1**, pero **preserva la forma** de la distribución original.

Propiedades Clave

- **Preserva la forma:** Una distribución normal seguirá siendo normal.
- **Comparación fácil:** Permite evaluar el peso de variables con distintas unidades.
- **Robustez moderada:** Es menos sensible a *outliers* que la normalización Min-Max.

Aplicaciones Comunes

- En **regresión**, para comparar la importancia de los coeficientes.
- Como paso previo a **PCA** o análisis discriminante.
- Cuando las variables tienen distribuciones aproximadamente simétricas.

Normalización Min-Max

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Esta técnica escala los datos a un rango fijo, comúnmente **[0, 1]**, donde 0 es el mínimo valor observado y 1 es el máximo.

- **Cuándo Usarlo:**
  - Algoritmos que requieren entradas en un rango específico, como las **redes neuronales**.
  - Cuando la interpretación en términos de mínimo y máximo es útil para el problema.
  - En datos con distribuciones uniformes o sin *outliers* extremos.
- **Limitación Principal:**
  - Es **muy sensible a *outliers***. Un solo valor extremo puede comprimir el resto de los datos en un rango muy pequeño, perdiendo información sobre su variabilidad.

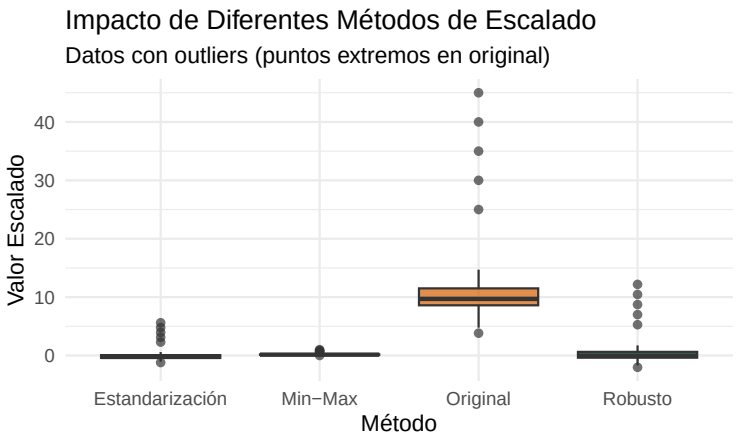
Escalado Robusto

$$X_{robust} = \frac{X - \text{mediana}(X)}{\text{IQR}(X)}$$

Diseñado específicamente para datos con *outliers*. Utiliza la **mediana** y el **rango intercuartílico (IQR)**, que son medidas estadísticas no afectadas por valores extremos.

- **Principio Clave:**
  - Al no usar la media ni la desviación estándar, los *outliers* no distorsionan el resultado del escalado.
- **Cuándo Usarlo:**
  - Es la opción preferida cuando se sabe o se sospecha que los **datos contienen *outliers*** significativos.
  - Cuando se quiere preservar la estructura de la mayor parte de los datos sin la influencia de los valores extremos.

Comparación de Métodos de Escalado



Observación: El escalado robusto mantiene mejor la estructura central ante outliers

Catálogo de Transformaciones según el Propósito

Una vez realizado el diagnóstico, debemos seleccionar la transformación más apropiada. La clave no está en *qué* transformación aplicar, sino en entender *por qué* esa transformación específica resuelve nuestro problema.

Exploraremos tres familias de transformaciones:

1. Para **linearizar relaciones** no lineales.
2. Para **estabilizar la varianza** (heterocedasticidad).
3. Para **normalizar residuos** y controlar *outliers*.

Transformación Logarítmica (Linearizar)

Es la transformación más versátil, ideal para linearizar relaciones de crecimiento exponencial o donde los efectos son multiplicativos.

Cuándo Usarla

- Relaciones **exponenciales** o **multiplicativas**.
- Procesos de **crecimiento proporcional**.
- Variables con **rendimientos decrecientes** (ingresos, precios).
- **Casos Típicos:** Economía, biología, finanzas.

Diagnóstico y Aplicación

- **Diagnóstico:** Curva cóncava que se aplana o varianza que aumenta con Y.
- **Aplicación:** `log(Y) ~ X`, `Y ~ log(X)` o `log(Y) ~ log(X)`.
- **Interpretación:** Los coeficientes se leen como cambios porcentuales o elasticidades.

Transformación de Potencia (Linealizar)

Fundamental para relaciones curvilíneas que siguen una **ley de potencia** del tipo `Y = a * X^b`.

- **Diagnóstico:** La relación entre las variables se vuelve lineal al graficarla en una **escala log-log**.
- **Aplicación:** Se toman **logaritmos en ambas variables** para linearizar el modelo: `log(Y) = log(a) + b * log(X)`.
- **Interpretación:** El exponente `b` representa la **elasticidad** o el exponente de escalamiento.
- **Ejemplos Clásicos:** Ley de Kleiber (relación masa-metabolismo), economía urbana (población-PIB).

Transformación Raíz Cuadrada (Estabilizar Varianza)

Especialmente útil para **datos de conteo** (típicamente de una distribución de Poisson), donde la varianza es proporcional a la media.

Cuándo Aplicarla

- **Conteos de eventos:** número de defectos, llamadas, ventas por período.
- **Datos de frecuencia:** visitas, clics, transacciones.
- Para conteos con muchos ceros puede requerir `sqrt(Y + 0.5)`.

Diagnóstico y Limitaciones

- **Diagnóstico:** Gráfico de residuos en forma de embudo donde la dispersión crece linealmente con la media.
- **Limitaciones:** La interpretación es menos directa y solo es apropiada para valores no negativos.

Transformación Logarítmica (Estabilizar Varianza)

Es la solución natural cuando la **varianza es proporcional al cuadrado de la media**, lo que se conoce como heterocedasticidad multiplicativa.

- **Cuándo Aplicarla:**
  - **Variables monetarias** (ingresos, precios, costos), donde el error relativo tiende a ser constante.
  - Porcentajes de crecimiento o procesos donde los **errores se acumulan multiplicativamente**.
- **La Gran Ventaja (Efectos Múltiples):**
  - Con frecuencia, esta única transformación resuelve varios problemas a la vez: **linealiza la relación, estabiliza la varianza, normaliza la distribución y reduce el impacto de *outliers***.

Transformación Inversa (Controlar Outliers)

Útil para relaciones **hiperbólicas** (del tipo `Y = 1/X`) y para distribuciones con colas muy pesadas a la derecha.

Cuándo Usarla

- Relaciones que se aproximan a una **asíntota horizontal**.
- **Tasas de decaimiento** o relaciones dosis-respuesta en farmacología.
- **Tiempo hasta un evento**.

Efecto y Precauciones

- **Efecto en *Outliers*:** Comprime fuertemente los valores grandes y expande los pequeños.
- **Precaución:** Amplifica errores en valores pequeños y requiere tratamiento especial para datos cercanos a cero.
- Solo aplicable a valores **no nulos**.

Transformación de Box-Cox: La Teoría

Es un método que **optimiza automáticamente** el parámetro de transformación λ (lambda) para maximizar la normalidad y homocedasticidad de los residuos. En lugar de elegir manualmente, Box-Cox encuentra el valor λ que mejor normaliza los datos.

Definición Matemática:

$$Y(\lambda) = \begin{cases} \frac{Y^{\lambda}-1}{\lambda}, & \lambda \neq 0 \\ \log(Y), & \lambda = 0 \end{cases}$$

Casos Especiales de λ:

- λ = **1:** Sin transformación (identidad).
- λ = **0.5:** Transformación de raíz cuadrada.
- λ = **0:** Transformación logarítmica.
- λ = **-1:** Transformación inversa.

Box-Cox: Propósito y Limitaciones

Propósito y Ventajas

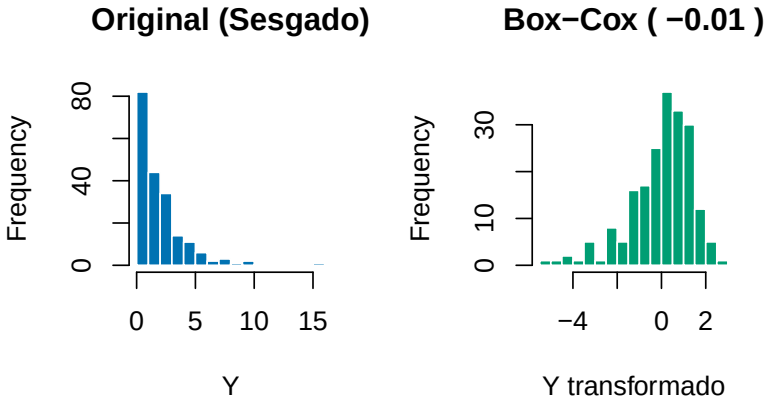
- Encuentra la transformación **óptima** sin necesidad de prueba y error.
- Maximiza la verosimilitud del modelo, mejorando simultáneamente la **normalidad** y la **homocedasticidad**.
- Proporciona un método **objetivo** para seleccionar la transformación más apropiada.

Limitaciones Importantes

- **Requiere que los datos (Y) sean estrictamente positivos**. Esta es su principal restricción.
- La **interpretación** es compleja si λ no es un valor simple (0, 0.5, 1).
- El λ óptimo **depende del modelo** específico, por lo que puede cambiar si se modifican los predictores.

Ejemplo Práctico: Aplicando Box-Cox

A continuación, se visualiza cómo la transformación Box-Cox corrige la asimetría de una distribución original.



La Extensión: Transformación de Yeo-Johnson

Fue desarrollada para superar la principal limitación de Box-Cox: **acepta cualquier valor real (positivo, negativo o cero)**.

Cuándo Usar Box-Cox

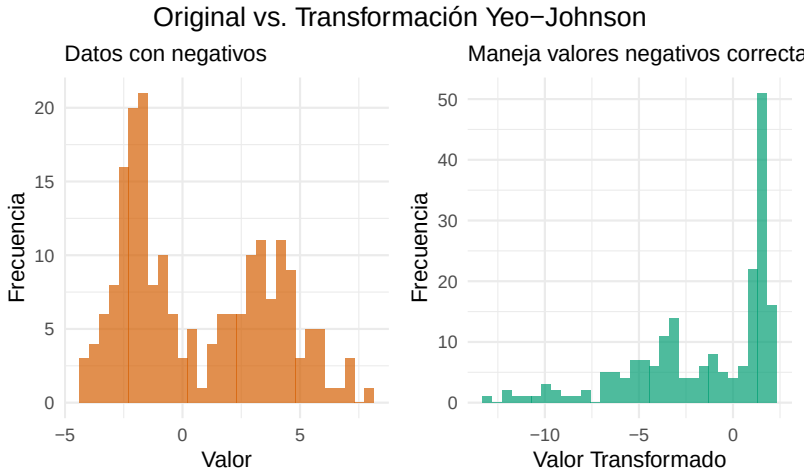
- Cuando los datos son **estrictamente positivos**.
- Si se busca comparabilidad con literatura existente que la utilice.

Cuándo Usar Yeo-Johnson

- Cuando los datos incluyen **valores negativos o cero**.
- Si se necesita mayor **flexibilidad** y no hay restricciones de dominio.

Ejemplo Práctico: Yeo-Johnson con Datos Negativos

Este ejemplo muestra cómo Yeo-Johnson maneja un conjunto de datos que incluye valores negativos, algo que Box-Cox no podría hacer.



Variables Categóricas: Principios de Codificación

La mayoría de los algoritmos estadísticos requieren entradas numéricas. El objetivo de la codificación es transformar las categorías de texto en números, **preservando su información semántica sin introducir supuestos erróneos**.

Criterios para Seleccionar el Método de Codificación:

- **Naturaleza de la variable:** ¿Existe un orden inherente entre las categorías? Esta es la pregunta más importante.
  - **Nominal:** Sin orden (ej. color, país).
  - **Ordinal:** Con orden (ej. nivel educativo, satisfacción).
- **Número de categorías:** Variables con muchas categorías ("alta cardinalidad") pueden requerir técnicas especiales.
- **Interpretabilidad del modelo:** ¿Qué método facilita una explicación clara de los resultados?

One-Hot Encoding (Variables Nominales)

Transforma una variable con **k categorías sin orden** (ej. color, región) en **k-1 variables binarias** (0/1), también conocidas como *variables dummy*.

La "Dummy Variable Trap":

- **Problema:** Usar `k` columnas (una para cada categoría) crea colinealidad perfecta en modelos lineales, ya que una columna es una combinación lineal de las otras.
- **Solución:** Siempre se omite una categoría, que se convierte en la **categoría de referencia** del modelo.

Ventajas

- No asume un orden entre categorías.
- Cada nueva variable tiene un coeficiente directamente interpretable (la diferencia con la categoría de referencia).

Desventajas

- Aumenta mucho la dimensionalidad si hay muchas categorías.
- Genera matrices de datos con muchos ceros (*dispersas*).

Ejemplo Práctico: One-Hot Encoding

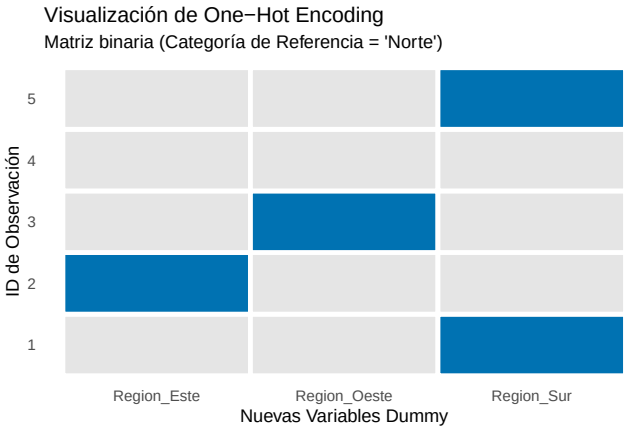
Imaginemos una variable `Region` con 4 categorías. Al codificarla, elegimos "Norte" como categoría de referencia.

Resultado:

- Se crean 3 variables binarias (k-1)
- Cada observación tiene un 1 en su región correspondiente
- "Norte" no aparece (categoría de referencia)

Interpretación:

El coeficiente de `Region_Sur` se interpretaría como la diferencia promedio en la variable respuesta al estar en la región "Sur" **en comparación con la región "Norte"**.



Codificación Ordinal (Variables Ordinales)

Asigna números enteros consecutivos a las categorías, **respetando su orden o jerarquía natural**.

Ejemplo Práctico:

- **Variable:** `Nivel_Satisfaccion` con categorías [“Bajo”, “Medio”, “Alto”].
- **Codificación:** Se asignan los valores [1, 2, 3].

Ventajas

- Preserva la información jerárquica de la variable.
- Es muy eficiente: crea una sola columna, sin importar el número de categorías.

Desventajas

- **Supone que la “distancia” entre niveles es uniforme** (asume que el cambio de “Bajo” a “Medio” es igual que de “Medio” a “Alto”).
- Puede imponer un orden artificial si se aplica por error a una variable nominal.

Comparación: Ordinal vs. One-Hot Encoding

La elección incorrecta del método puede llevar a modelos con menor rendimiento e interpretaciones erróneas.

Usar Codificación Ordinal

- **Cuándo:** Para variables con un **orden natural y significativo** (ej. nivel educativo, satisfacción del cliente, grado de severidad).
- **Resultado:** Una sola columna numérica (`1, 2, 3, ...`).
- **Riesgo Principal:** Asumir que los intervalos entre categorías son iguales.

Usar One-Hot Encoding

- **Cuándo:** Para variables **nominales**, sin un orden inherente (ej. color, género, país).
- **Resultado:** `k-1` columnas binarias (`0` o `1`).
- **Riesgo Principal:** Aumento excesivo del número de variables si la cardinalidad es alta.

¿Qué es una Interacción?

Mientras los efectos principales miden el impacto promedio de una variable, las **interacciones** revelan cómo el efecto de una variable **cambia según el nivel de otra**.

Modelo con Interacción:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 \times X_2 + \varepsilon$$

Interpretación de β_3:

- Si $\beta_3 > 0$: Los efectos se **potencian** (sinergia).
- Si $\beta_3 < 0$: Los efectos se **atenúan** (compensación).

Ejemplo Visual: Interacción Experiencia × Género

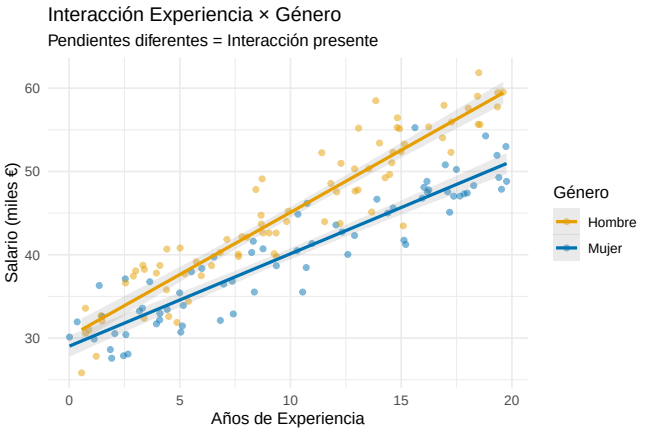
Un ejemplo clásico es cómo la relación entre experiencia y salario no es la misma para todos los grupos.

¿Qué observamos?

- **Pendientes diferentes**: Cada género tiene una relación distinta entre experiencia y salario
- **Interacción presente**: El efecto de la experiencia depende del género

Interpretación:

La interacción permite que la pendiente de la experiencia sea diferente para cada género, revelando patrones que el análisis por separado no detectaría.



Tipos de Interacciones: Continua × Continua

Este tipo de interacción ocurre cuando el efecto de una variable continua sobre el resultado depende del valor de otra variable continua.

- **Concepto:** El efecto de una variable se **amplifica** o **atenúa** a medida que otra variable cambia.
- **Modelo:** $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 \times X_2$. El coeficiente β_3 captura la interacción.

Sintaxis en R:

```
modelo <- lm(ventas ~ precio * publicidad, data = datos)
# Interpretación: El efecto del precio sobre las ventas varía
# según el nivel de inversión en publicidad.
```

Tipos de Interacciones: Categórica × Categórica

Ocurre cuando el efecto de pertenecer a una categoría depende de la pertenencia a otra categoría.

- **Concepto:** Existe un **efecto específico para una combinación de categorías** que no se puede explicar sumando los efectos individuales.
- **Ejemplo:** El efecto del género en el salario puede ser diferente en cada departamento de una empresa.

Sintaxis en R:

```
modelo <- lm(salario ~ genero * departamento, data = datos)
# Interpretación: La brecha salarial de género es diferente
# en cada departamento.
```

Tipos de Interacciones: Continua × Categórica (Mixta)

Este es uno de los tipos más comunes e intuitivos. Permite que la relación entre una variable continua y el resultado sea diferente para distintos grupos.

- **Concepto:** La **pendiente** de la variable continua es diferente para cada nivel de la variable categórica.
- **Ejemplo:** La relación entre los años de experiencia y el salario puede tener una pendiente más pronunciada para un grupo que para otro.

Sintaxis en R:

```
modelo <- lm(rendimiento ~ horas_estudio * metodo, data = datos)
# Interpretación: La efectividad de las horas de estudio
# (la pendiente)
# sobre el rendimiento varía según el método de estudio
# utilizado.
```

Caso Práctico: Visualizando una Interacción Categórica

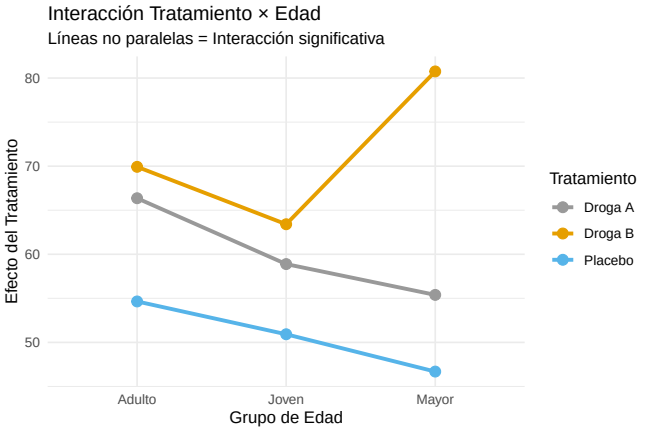
Los *interaction plots* son la mejor herramienta para entender interacciones entre variables categóricas. **Las líneas no paralelas son una señal visual clara de una posible interacción**.

¿Qué buscamos?

- **Líneas paralelas**: Sin interacción
- **Líneas no paralelas**: Interacción presente
- **Líneas que se cruzan**: Interacción fuerte

Interpretación:

La Droga B es especialmente efectiva en el grupo de "Mayor", un efecto que no se podría ver analizando las variables por separado.



Detección y Estrategia de Modelado

No debemos buscar interacciones al azar. El enfoque correcto combina la teoría con la evidencia de los datos.

¿Cómo Detectarlas?

- **Justificación Teórica:** El conocimiento del dominio es la guía principal para saber dónde buscar.
- **Exploración Visual:** Usar gráficos de dispersión por grupos o *interaction plots*.
- **Tests Estadísticos:** Utilizar el Test F para comparar formalmente un modelo con y sin la interacción.

El Principio de Jerarquía

- **Regla de Oro:** Si incluyes una interacción `A × B`, **siempre debes incluir los efectos principales `A` y `B` por separado**.
- **Justificación:** Preserva la interpretabilidad del modelo y evita sesgos en los coeficientes.

Riesgos y Consideraciones Prácticas

Las interacciones son poderosas, pero su uso requiere cuidado para no complicar el modelo innecesariamente.

- **Riesgo de Complejidad:**
  - El número de interacciones posibles crece exponencialmente.
  - **Recomendación:** Limitar el modelo a las 2 o 3 interacciones más importantes y con justificación teórica.
- **Riesgo de Multicolinealidad:**
  - Las interacciones pueden hacer que los coeficientes sean inestables.
  - **Mitigación:** Centrar las variables continuas antes de crear el término de interacción.
- **Interpretación con Transformaciones:**
  - **Advertencia:** Una interacción en una escala `log` no significa lo mismo que en una escala lineal y requiere una interpretación mucho más cuidadosa.

Consiste en crear nuevas variables mediante **combinaciones, ratios y transformaciones compuestas**. El objetivo es capturar relaciones complejas que no son evidentes en las variables originales.

A menudo, las variables individuales contienen información parcial. Al combinarlas de forma inteligente, podemos revelar **patrones predictivos mucho más potentes**.

Técnicas Clave que Exploraremos:

- **Combinaciones:** Creación de nuevas variables a partir de sumas o productos de las existentes.
- **Ratios y Proporciones:** Normalización de variables para revelar relaciones estructurales.
- **Manejo de Colinealidad:** Estrategias para condensar información de predictores correlacionados.

Creación de Features por Combinación

Combinaciones Lineales

- **Concepto:** Sumas ponderadas de variables que miden aspectos de un mismo fenómeno.
- **Ejemplo (Índice Compuesto):** Un índice de riesgo cardiovascular. `Índice_Riesgo = 0.4*Presión + 0.3*Colesterol + 0.3*IMC`
- **Aplicación:** Crear un *score* único a partir de múltiples indicadores para capturar un constructo multidimensional.

Combinaciones No Lineales

- **Concepto:** Productos, cocientes o funciones complejas que capturan sinergias o efectos multiplicativos.
- **Ejemplo (Producto de Eficiencia):** `Rendimiento = Capacidad × Utilización × Calidad`
- **Aplicación:** Modelar efectos donde el resultado depende de la combinación simultánea de varios factores.

El Poder de los Ratios

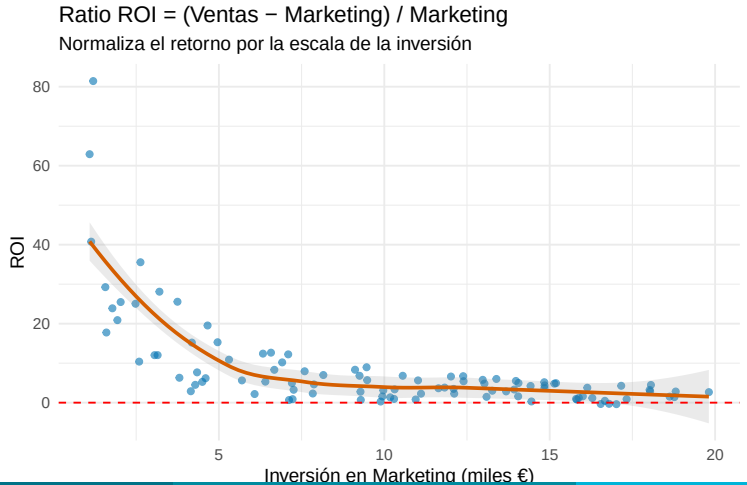
Los ratios son muy potentes porque **normalizan automáticamente las diferencias de escala** y revelan relaciones estructurales que las variables absolutas ocultan.

Ventajas Principales:

- **Normalización Automática:** Permiten comparar entidades de diferentes tamaños (ej. una startup vs. una multinacional).
- **Interpretación Intuitiva:** Tienen significados claros y directos (ej. `Deuda / Patrimonio`).
- **Robustez:** Suelen ser menos sensibles a valores atípicos (*outliers*).

Ejemplo Práctico: Retorno de Inversión (ROI)

Un ratio clásico en negocio es el ROI, que mide la eficiencia de una inversión normalizando el beneficio obtenido por el coste de la misma.



Manejo de Colinealidad con Features

El Problema: Simplemente eliminar variables correlacionadas es una mala práctica, ya que **se pierde información predictiva valiosa**.

La Solución: Condensar la información redundante en nuevas variables, **preservando la información única** de cada predictor original.

Estrategias Principales

- **Componentes Principales (PCA):** Extrae la máxima varianza común en componentes ortogonales. Su principal desventaja es que **pierde interpretabilidad directa**.
- **Ratios Informativos:** Capturan la relación estructural entre dos variables (ej. `Deuda / Patrimonio`). **Preservan la interpretabilidad económica**.
- **Índices Ponderados:** Crean un *score* único a partir de varias variables, usando pesos teóricos o empíricos.

El Trade-Off: Complejidad vs. Interpretabilidad

A medida que aumenta la complejidad de la técnica, generalmente se gana poder predictivo pero se pierde facilidad de interpretación.

Técnicas Más Simples

- **Estandarización**: La pérdida de interpretación es **mínima** (solo cambia la escala).
- **Transformación Logarítmica**: La pérdida es **baja**, ya que se puede interpretar como cambios porcentuales.

Técnicas Más Potentes

- **Box-Cox (con λ complejo)**: La escala transformada no es intuitiva, dificultando la interpretación.
- **PCA (Componentes Principales)**: La pérdida es **muy alta**, ya que los componentes son constructos abstractos.
- **Interacciones Múltiples**: La interpretación se complica al haber **efectos condicionales**.

- **Si buscas EXPLICAR** $\rightarrow$ Prioriza la **Interpretabilidad**.
- **Si buscas PREDECIR** $\rightarrow$ Prioriza el **Rendimiento**.

Errores Comunes a Evitar

1. **Transformar sin Diagnóstico Previo**
   - Aplicar `log()` "por si acaso" en lugar de identificar un problema específico que lo justifique.
2. **Ignorar el Dominio del Problema**
   - Usar transformaciones que no tienen sentido teórico (ej. `log(edad)`).
3. **Caer en la "Dummy Variable Trap"**
   - Incluir `k` columnas para `k` categorías en lugar de `k-1` (categoría de referencia).
4. **Usar Interacciones sin Efectos Principales**
   - Modelar `Y ~ A:B` en lugar de la forma correcta y jerárquica `Y ~ A + B + A:B`.
5. **Sobreingeniería (*Over-engineering*)**
   - Crear cientos de *features* automáticamente sin una justificación clara y validada para cada uno.

Flujo de Trabajo Recomendado

Un proceso sistemático garantiza resultados robustos y reproducibles.

1. **Análisis Exploratorio**
   - Visualizar distribuciones, patrones y *outliers*.
2. **Diagnóstico de Problemas**
   - Buscar no linealidad, heterocedasticidad, asimetría, etc., en los datos.
3. **Selección de Transformaciones**
   - Elegir la técnica adecuada para el problema diagnosticado.
4. **Aplicación y Validación**
   - Transformar las variables y verificar si el problema original se ha resuelto.
5. **Comparación de Modelos**
   - Usar métricas (R^2, RMSE, AIC) y validación cruzada para evaluar la mejora.
6. **Interpretación Final**
   - Traducir los resultados del modelo final al contexto del negocio o la investigación.

Mejores Prácticas en Feature Engineering

1. **Documentación Rigurosa**
   - Registrar cada transformación y su justificación para mantener la trazabilidad.
2. **Validación en Datos Nuevos**
   - Guardar los parámetros de transformación del set de entrenamiento (ej. media, sd, lambda) y aplicarlos al de test.
3. **Evitar *Data Leakage***
   - Calcular todos los parámetros de las transformaciones **únicamente con los datos de entrenamiento**.
4. **Considerar el Contexto**
   - Asegurarse de que las nuevas variables creadas son interpretables.
5. **Parsimonia**
   - Preferir siempre el modelo más simple que funcione adecuadamente. No añadir complejidad por mejoras marginales.

Resumen y Conceptos Clave

¿Por Qué es Fundamental?

- **Resuelve problemas** específicos de los datos (no linealidad, etc.).
- **Mejora el rendimiento** del modelo significativamente.
- **Revela relaciones ocultas** mediante interacciones y combinaciones.
- **Adapta los datos** a los requisitos de los algoritmos.

Principios Clave

- **Diagnóstico** antes que transformación.
- **Justificación** teórica o empírica.
- **Validación** rigurosa.
- **Balance** entre complejidad e interpretabilidad.
- **Documentación** exhaustiva.

Próximo tema: Selección de variables, regularización y validación para manejar la complejidad agregada.