Variables categóricas

- Ahora en mi DataFrame además de variables numéricas, tengo variables categóricas
- Las categorías NO se pueden ordenar, ni sumar, ni restar…
- Ejemplo:
  - Columna ‘País’: ‘España’, ‘México’, ‘Portugal’, ’España’…
  - Columna ‘Color’: ‘Verde’, ‘Rojo’, ‘Azul’, ‘Rojo’, ‘Gris’…

  - No se ordenan numéricamente porque no se puede decir que una categoría sea mayor que otra
  - Atención: existen categorías cuya etiqueta es un número, pero siguen siendo categorías. Ejemplo: DNI sin letra, se refiere a una persona, un DNI con un número alto NO vale más que un DNI con un valor bajo

	País_último_viaje	DNI	Teléfono	Color_preferido
0	España	1234	98765	rojo
1	Portugal	2345	87654	verde
2	España	3456	76543	azul
3	Francia	4567	65432	azul
4	España	5678	54321	verde
5	Portugal	6789	43219	azul
6	Portugal	7891	32198	rosa
7	Francia	8912	21987	morado
8	España	9123	19876	gris

```
df_categorias.dtypes
```

```
País_último_viaje    object
DNI                   int64
Teléfono              int64
Color_preferido      object
dtype: object
```

Ojo: pandas por defecto entiende que un número es de tipo numérico. Si un número es una categoría, tenemos que decírselo nosotros explícitamente. Los strings son de tipo objeto.

```
df_categorias.DNI = (df_categorias.DNI).astype(str)
df_categorias.DNI.dtypes
```

```
dtype('O')
```

Ahora en el DNI hemos cambiado el tipo de int64 a str.

Vemos que ya es de tipo O (objeto)

Codificación de categorías

- Un codificador lo que hace es asociar una categoría a un número, respetando **NO darle más peso a una etiqueta que a otra** (si las etiquetas no lo tienen; “primero”, ”segundo”, “tercero” sí tienen un orden y puedo codificar 1, 2, 3)
- Vamos a aprender 5 formas distintas en las que se puede codificar:
  1. One-hot
  2. Dummy
  3. Effect
  4. Hash
  5. Estadístico de etiqueta
- Nos interesa manejar números porque los ordenadores trabajan muy bien con números

1. One-Hot encoder

- Cada valor distinto dentro de la columna se convierte en una nueva columna binaria

```
pd.get_dummies(df_categorias.País_último_viaje, prefix='Viaje')
```

	País_último_viaje
0	España
1	Portugal
2	España
3	Francia
4	España
5	Portugal
6	Portugal
7	Francia
8	España

	Viaje_España	Viaje_Francia	Viaje_Portugal
0	1	0	0
1	0	0	1
2	1	0	0
3	0	1	0
4	1	0	0
5	0	0	1
6	0	0	1
7	0	1	0
8	1	0	0

1. One-Hot encoder

- Podría colocar las columnas Viaje_España, Viaje_Francia y Viaje_Portugal en el orden que quisiera (España-Francia-Portugal o Francia-Portugal-España o Portugal-Francia-España…) nota que cada país **SOLO AFECTA A UNA DE LAS COLUMNAS**
- De este modo, poniendo a UNO solo a una columna y las demás a CERO, NO estoy diciendo que un país valga más que otro, solo estoy marcando en esas columnas desplegadas el valor de la fila en la columna con las etiquetas
- ¡Acabamos de codificar etiquetas de forma numérica y sin ponderar más unas etiquetas que otras!

INCONVENIENTE (¡PELIGRO!, ¡PELIGRO!)

DUMMY VARIABLE TRAP

Dummy variable trap
(la trampa de la variable dummy)

- En One-hot encoder, si utilizo todos los valores únicos de la etiqueta (España, Francia, Portugal) estoy utilizando una columna REDUNDANTE, porque siempre hay una columna que se puede obtener como combinación lineal del resto.

- Ejemplo:
  - Portugal es 1→ si no es España ni Francia (si España y Francia son 0)
  - Francia → si no es Portugal ni España (si Portugal y España son 0)
  - España → si no es Francia ni Portugal (si Francia y Portugal son 0)

Dummy variable trap

- Ejemplo:
  - Portugal = NOT(España OR Francia)

```
Portugal = (one_hot.Viaje_España | one_hot.Viaje_Francia).apply(lambda x: not x)
```

```
   Portugal
0     False
1      True
2     False
3     False
4     False
5      True
6      True
7     False
8     False
dtype: bool
```

Viaje_Portugal
0
1
0
0
0
1
1
0
0

One-hot encoder SIEMPRE tiene una columna que se puede obtener del resto por **combinación lineal**. Es decir, One-hot encoder SIEMPRE tiene una columna que NO aporta información.

Me interesa que las columnas NO SEAN combinaciones lineales de otras.

Si las columnas son combinaciones lineales → las soluciones de mi modelo NO SERÁN ÚNICAS

Si EVITO LA COLINEALIDAD entre columnas → la solución de mi modelo ES ÚNICA

Dummy trap variable

Mi objetivo→ y_predicción = función(W0 + W1 x1 + W2 x2 + W3 x3 + W4 x4)
Con X_train, Y_train obtengo los parámetros de mi modelo (W0,W1,W2,W3,W4)
Con X_test, Y_test testeo con datos intactos cómo de bueno es mi modelo

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
0	5.1	3.5	1.4	0.2	0
1	4.9	3	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5	3.6	1.4	0.2	0
5	5.4	3.9	1.7	0.4	0
6	4.6	3.4	1.4	0.3	0
7	5	3.4	1.5	0.2	0
8	4.4	2.9	1.4	0.2	0
9	4.9	3.1	1.5	0.1	0

X_train
X_test
Y_train
Y_test

Dummy trap variable

Por sencillez de lectura uso a,b y c:

$y = W0 + W1\ x1 + W2\ x2 + W3\ x3$
$c = W0 + W1\ a + W2\ b$

- Supongamos que tengo dos variables, A y B, donde A es género masculino y B es género femenino. Claramente A+B=1 y B=1-A
- Quiero predecir c a partir de (a,b)
- $c = W0 + W1\ a + W2\ b$
- Mi modelo consiste en descubrir W0, W1 y W2
- COLINEALIDAD: A y B son columnas COLINEALES
- $c = W0 + W1\ a + W2\ (1-a) = W0+W2+ (W1-W2)\ a$
- $c = W0 + W1(1-b) + W2\ b = W0 + W1 – W1b + W2b = W0 + W1+(W2-W1)b$
- (W0,W1,W2) = (1,2,3) = (4,-1,0) = (3,0,1) Cualquiera de esos pesos son una solución

2. Dummy encoder

- Como el One-Hot encoder pero **ELIMINANDO una columna**

```
dummy_df = pd.get_dummies(df_categorias.País_último_viaje, prefix='Viaje',  drop_first=True)
```

	País_último_viaje
0	España
1	Portugal
2	España
3	Francia
4	España
5	Portugal
6	Portugal
7	Francia
8	España

	Viaje_Francia	Viaje_Portugal
0	0	0
1	0	1
2	0	0
3	1	0
4	0	0
5	0	1
6	0	1
7	1	0
8	0	0

- Esta codificación se usa MUCHO con categorías.
- Usamos Dummy, One-hot generalmente NO
- **get_dummies NO te quita una columna por defecto, hay que decírselo**
- Recuerda: eliminar una columna TE AYUDA, no es ningún problema, ¡te evita problemas!

3. Effect encoder

- Cuando utilizamos un Dummy encoder y eliminamos una columna, lo que realmente estamos haciendo es poner a 0 una columna codificada entera.
- Si tuviéramos una columna que tuviese todos sus valores a 0, el Dummy encoder pensaría que es esa columna que hemos eliminado (la eliminada de referencia)
- Para poder distinguir entonces entre el caso de una columna que sean todo 0 y la columna eliminada, el Effect encoder lo que hace es poner a -1 todos los valores de la columna de referencia. Ahora eliminamos una columna para tener soluciones únicas y además podemos distinguir si nos encontramos con una columna nula entera.
- El problema es que almacenar vectores muy grandes con -1 es mucho más costoso computacionalmente que poniendo la columna eliminada a 0
- Pandas NO USA Effect encoder, usa One-hot y Dummy si tú se lo indicas explícitamente

4. Hash encoder

- Cuando tenemos muchas etiquetas distintas en una columna (ahora no son 3 países, pensemos en gente haciendo click en anuncios de webs donde en el servidor final se guarda desde dónde se hizo click (dirección IP) origen y en qué anuncio.
- La combinación de IP y anuncios de millones de usuarios es MUY GRANDE
- Si usamos un Dummy, esa columna nos generaría millones de columnas (menos una), una por cada par usuario-anuncio abierto (¡menos un usuario-anuncio, que es un Dummy encoder!
- Esto serían demasiadas columnas nuevas
- Las funciones Hash son funciones matemáticas que de un espacio de entrada muy grande, mapean en un espacio de salida mucho más reducido
- Evidentemente varias entradas se mapean en el mismo valor de salida
- Siempre que introduces una entrada, te genera la misma salida

5. Estadístico de etiqueta

- Otra forma de trabajar cuando tenemos MUCHAS etiquetas es asociando algún estadístico a esa etiqueta.
- Así de UNA etiqueta generamos UN NÚMERO
- Tenemos que recordar primero un poco de…
  - PROBABILIDAD CONDICIONADA
  - Tenemos dos usuarios: Juan y Ana; ambos hacen click en anuncios.
  - P(click|Juan)→probabilidad de hacer click si SÉ QUE ES JUAN
  - P(click|Ana)→probabilidad de hacer click si SÉ QUE ES ANA
  - P(Ana|click)→probabilidad de ser Ana si SÉ QUE HA HECHO CLICK
  - P(Juan|click)→probabilidad de ser Juan si SÉ QUE HA HECHO CLICK

5. Estadístico de etiqueta

- Teorema de Bayes (columnas x independientes)
- P(Clase|x) = (P(Clase) P(x|Clase)) / sum clases (P(Clase) P(x|Clase))

- P(Clase|x) → probabilidad a posteriori
- P(x|Clase) → probabilidad a priori
- P(Clase) → probabilidad de clase
- sum clases (P(x|Clase) P(Clase)) → constante normalización

- P(click|Juan)→probabilidad a priori Clase = Juan
- P(click|Ana)→probabilidad a priori Clase = Ana
- P(Ana|click)→probabilidad a posteriori Clase = Ana
- P(Juan|click)→probabilidad a posteriori Clase = Juan

5. Estadístico de etiqueta

	usuario	clicks_dados	anuncios_vistos
0	Juan	7	28
1	Ana	3	19

- Clases = Juan, Ana
- x = hacer click

- P(Juan) = ½
- P(Ana) = ½
- P(click|Juan) = 7 / 28 = 0,25
- P(click|Ana) = 3/19 = 0,16

5. Estadístico de etiqueta

	usuario	clicks_dados	anuncios_vistos
0	Juan	7	28
1	Ana	3	19

- Naive Bayes (suponemos los atributos independientes y la constante de normalización se calcula entonces como una suma del numerador aplicado en cada clase)

- P(Juan|click) = (0,25 x ½) / (0,25 x ½ + 0,16 x ½) = 0,61
- P(Ana|click) = (0,16 x ½) / (0,25 x ½ + 0,16 x ½) = 0,39

5. Estadístico de etiqueta

- Ahora puedo añadir columnas con estadísticos que dependen de las etiquetas (‘Juan’, y ‘Ana’), pero se genera una columna por estadístico.
- Supongamos que tengo 1000 nombres, NO se van a generar 999 columnas como con el Dummy, si incluyo la probabilidad a priori se me genera UNA COLUMNA con mil filas, si añado la probabilidad a posteriori, se me genera OTRA columna con mil filas, si añado la probabilidad de clase, se me genera OTRA columna con mil filas…
- Ya no me aparecen muchas columnas por tener muchas etiquetas
- Añadiré los estadísticos que vienen de las etiquetas que me interesen (existen más estadísticos), evitando después las propias etiquetas.