ANÁLISIS DE DATOS

Clase 5. Taller de preparación de datos (final)

Temario

1. Encoding de variables categóricas
2. Cadenas de procesamiento.
3. Conceptos básicos de Teoría de la información
   - Entropía y entropía conjunta
   - Entropía relativa
   - Información mutua
4. Test estadísticos

CODIFICACIÓN DE VARIABLES CATEGÓRICAS

Codificación de variables categóricas.

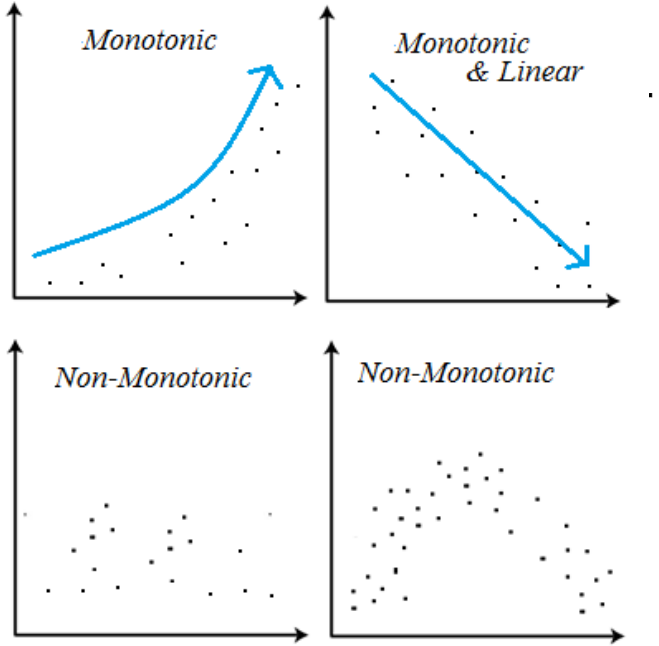
- Consiste en reemplazar una categoría (típicamente representada en forma de texto) por una representación numérica.
- El objetivo es disponer de variables que puedan ser utilizadas en los modelos de AA.

Codificación de variables categóricas. Técnicas.

Tradicionales	• One Hot Encoding • Count/frequency encoding • Ordinal/label encoding
Relación monotónica	• Label encoding ordenado. • Encoding por promedio • Peso de la evidencia (WoE)
Alternativas	• Binary encoding • Feature hashing • Otros

Relación monotónica

- Decimos que existe una relación monotónica entre una variable independiente X e una variable objetivo Y cuando:
  - Si se incrementa el valor de X ta de Y ó
  - Si se incrementa el valor de X, Y



Relación monotónica. Importancia.

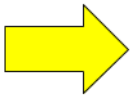
- Las relaciones monotónicas entre variables de entrada y de salida pueden mejorar el desempeño de los modelos lineales.
- En los modelos de árboles, puede traducirse en menor profundidad (sin comprometer su desempeño).
- A menudo, estas relaciones aparecen de manera natural en los datos. Por ejemplo: las primas de los seguros suelen disminuir a medida que aumenta la edad de los asegurados.
- Algunas de las formas de codificación que se mostrarán, estarán orientadas intentar obtener esta relación entre la variable transformada y la variable objetivo.

One Hot Encoding

- Consiste en codificar cada valor de una variable categórica con un conjunto de variables booleanas que pueden tomar 0 o 1, indicando si esa categoría está o no presente en cada observación.
- Si la variable tiene k-valores posibles, puede codificarse utilizando k o k-1 nuevas variables (en el último caso se suele llamar dummy encoding).

Color
Red
Red
Yellow
Green
Yellow

OHE



Red	Yellow	Green
1	0	0
1	0	0
0	1	0
0	0	1

DE

	Travel_Class_2	Travel_Class_3
Passenger 1	0	0
Passenger 2	1	0
Passenger 3	0	1
Passenger 4	0	0

One Hot Encoding. k o k-1?

- Codificar utilizando k-1 variables disminuye el costo adicional de creación de variables (tener presente que es común realizar entrenamientos sobre el dataset completo).
- En algunos modelos que tienen un término de sesgo (bias), como por ejemplo regresión lineal, la presencia de una matriz OHE con k variables haría que la matriz sea singular y por lo tanto no invertible, imposibilitando la solución por fórmula cerrada.
- No obstante, para los siguientes escenarios se aconseja utilizar k variables:
  - Algoritmos basados en árboles.
  - Feature selection por algoritmos recursivos.
  - Para determinar la importancia de cada categoría.

One Hot Encoding. Ventajas.

- No realiza ningún supuesto sobre la distribución de las categorías de la variable.
- Mantiene toda la información de la variable categórica.
- Es apta para modelos lineales.

One Hot Encoding. Limitaciones.

- Aumenta la dimensión del espacio de variables de entrada.
- No agrega información.
- Introduce muchas variables con información redundante.

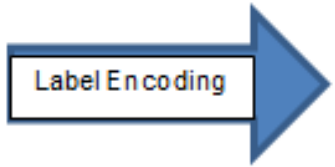
One Hot Encoding. Variante para top-n categorías.

- Una variante de OHE es utilizarla sólo para las top n categorías más frecuentes.
- Esta técnica fue la solución ganadora en KDD 2009: http://www.mtome.com/Publications/CiML/CiML-v3-book.pdf

Label/integer encoding. Definición.

- Consiste en reemplazar las categorías por valores numéricos de 0 a N-1.
- Los números se asignan de manera arbitraria.

	occupation
0	programmer
1	data scientist
2	engineer
3	manager
4	ceo



	occupation
0	4
1	1
2	2
3	3
4	0

Label/integer encoding. Ventajas.

- Fácil de implementar, no expande la dimensión del espacio de variables de entrada.
- Puede funcionar bien con algoritmos basados en árboles.

Label/integer encoding. Limitaciones.

- La codificación no aporta información.
- No es apropiado para modelos lineales.
- En producción, no maneja automáticamente nuevas categorías.

Count/Frequency encoding. Definición.

- Cada categoría se reemplaza por el valor o porcentaje de observaciones en que aparece en el dataset.
- Se captura la representación de cada categoría.
- Muy utilizado en competencias de Kaggle.
- Se hace un supuesto importante: la cantidad de observaciones para cada categoría está relacionada con la variable a predecir.

Count/Frequency encoding. Ventajas.

- Fácil de implementar.
- No expande las dimensiones del espacio de variables de entrada.
- Puede tener un desempeño aceptable con modelos basados en árboles.

Count/Frequency encoding. Limitaciones.

- No apropiado para modelos lineales.
- No maneja automáticamente nuevas categorías en test set/producción.
- Si aparecen dos categorías la misma cantidad de veces, pueden ser reemplazadas por el mismo número, con la consecuente pérdida de información.

Ordinal encoding c/ orden. Definición

- Consiste en reemplazar las categorías por valores numéricos de 0 a N-1, pero en este caso el ordenamiento no es arbitrario.
- La asignación de cada entero asignado corresponde con el promedio de la variable objetivo de cada categoría.

Ordinal encoding c/ orden. Ventajas

- Fácil de implementar.
- No expande las dimensiones del espacio de variables de entrada.
- Crea una relación monotónica entre las categorías y la variable objetivo.

Ordinal encoding c/ orden. Limitaciones

- Puede introducir overfitting.
- Las librerías estándar como SkLearn no lo soportan directamente, por lo que no es directo su uso en un esquema de cross-validation o k-folds validation.

Mean encoding. Definición

- Consiste en reemplazar cada categoría por el promedio de la variable objetivo para esa categoría.

Mean encoding. Ventajas

- Fácil de implementar.
- No expande las dimensiones del espacio de características.
- Crea una relación monotónica entre las categorías y la variable objetivo.

Mean encoding. Limitaciones

- Puede introducir overfitting.
- Las librerías estándar como SkLearn no lo soportan directamente, por lo que no es directo su uso en un esquema de cross-validation o k-folds validation.
- Puede ocurrir un escenario en el que dos categorías tengan un promedio muy similar, con la consecuente pérdida de información.

Peso de Evidencia (*Weight of Evidence*).

Definicion

- Esta técnica se originó para modelos financieros y riesgo crediticio.
- Consiste en reemplazar una variable binaria por el logaritmo natural del ratio del valor positivo respecto del valor negativo o por defecto (el orden puede invertirse dependiendo del caso, por ejemplo, en modelos financieros se suele usar p(0)/p(1)).

$$WoE = ln\frac{P(X=1)}{P(X=0)}$$

Peso de Evidencia (*Weight of Evidence*).

Ventajas.

- Crea una relación monotónica entre la variable objetivo y las variables de entrada.
- Funciona muy bien con regresión logística, por la forma en que quedan ordenadas las categorías.
- Las variables transformadas pueden ser comparadas porque están en la misma escala, por lo tanto, es casi inmediato determinar cuál es más predictiva.

Peso de Evidencia (Weight of Evidence). Limitaciones.

- Puede llevar a overfitting.
- Indefinida cuando el denominador es cero.

Codificación de etiquetas poco frecuentes.

- Las etiquetas poco frecuentes aparecen en una proporción pequeña de las observaciones del dataset.
- Suelen presentarse en los siguientes escenarios:
    - Variables con una categoría predominante.
    - Variables con pocas categorías.
    - Variables con alta cardinalidad.
- La recomendación es agrupar todas las categorías poco frecuentes en una nueva categoría “raras”.

Binary encoding / feature hashing. Definición

- Binary encoding es una combinación de OHE y ordinal encoding. Se codifica cada variable de entrada como una composición de variables binarias.
- Feature hashing, aplica una función de hash a cada valor de entrada para obtener un entero.
- En ambos casos:
  - Ventaja → eficiencia en la representación
  - Desventaja → pérdida de interpretabilidad.

			Binary Encoded			
Categorical Feature		=	x1	x2	x4	x8
Louise	=>	1	1	0	0	0
Gabriel	=>	2	0	1	0	0
Emma	=>	3	1	1	0	0
Adam	=>	4	0	0	1	0
Alice	=>	5	1	0	1	0
Raphael	=>	6	0	1	1	0
Chloe	=>	7	1	1	1	0
Louis	=>	8	0	0	0	1
Jeanne	=>	9	1	0	0	1
Arthur	=>	10	0	1	0	1

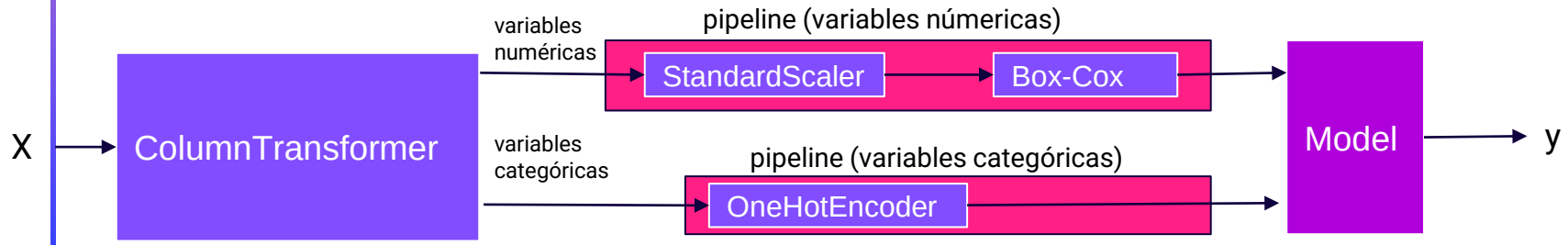
Ejemplos en jupyter

Clase 4.4 - Preparación de datos - Codificación de variables categóricas.ipynb

CADENAS DE PROCESAMIENTO

Implementación de cadenas de procesamiento con SKLearn

- Arquitectura de SKlearn:
  - **Transformer**: clase para transformar datos. Debe implementar fit() y transform().
  - **Predictor:** clase para realizar predicciones. Debe implementar fit() y predict().
  - **Pipeline:** clase que ejecuta secuencia de transformers y/o predictors.
    - Cada elemento de la lista es un paso (step).
    - El único elemento de la lista que puede ser predictor es el último (los predecesores deben ser transformers).
  - **Columntransformer**: clase que permite aplicar transformaciones específicas para cada columna de un dataset.



Implementación de cadenas de procesamiento con SKLearn

- Ejemplos de uso de clases **Pipeline** y **ColumnTransformer** para problemas de clasificación y regresión.
- Referencias:
  - https://scikit-learn.org/stable/modules/generated/sklearn.pipeline.Pipeline.html
  - https://scikit-learn.org/stable/modules/compose.html#columntransformer-for-heterogeneous-data
  - Sección de trransformaciones de datos de documentación de SKLearn: https://scikit-learn.org/stable/data_transforms.html

Ejemplos en jupyter

Clase 4.8 - Preparación de datos - Cadenas de procesamiento.ipynb

Resumen

El objetivo de estas clases fue presentar algunas de las técnicas más utilizadas para la preparación de datasets y cómo combinarlas para construir cadenas de procesamiento en SKLearn.

En las siguientes clases, se estudiarán otros aspecto de la **preparación de datos**:

- La **selección de variables** de entrada: ¿Qué variables son más relevantes para el modelo de AA?
- **Ingeniería de variables** (como generar variables que aporten mayor información, a partir de las existentes).
- **Reducción de dimensiones**. Creación de un nuevo espacio de variables entrada, a partir de proyecciones compactas.

Bibliografía y referencias

- *"Python Feature Engineering Cookbook" .**Soledad Galli.** Packt (2020).*
- *"Applied Predictive Modeling". **Max Kuhn; Kjell Johnson**. Springer (2016).*
- *"Feature Engineering and Selection". **Max Kuhn; Kjell Johnson**. CRC Press (2016).*
- *"Feature Selection for Data and Pattern Recognition". **Urszula Stańczyk; Lakhmi C. Jain**. Springer (2014).*
- *"Feature Engineering for Machine Learning". **Alice Zheng; Amanda Casari**. O'Reilly (2018).*
- *"The Art of Feature Engineering". **Pablo Doboue**. Cambridge University Press (2020).*
- *"Feature Engineering IFT6758 - Data Science"* Université de Montréal.

DUDAS?

ENCUESTA