Universidad Popular del Cesar

Procesamiento de Señales II
Ciencia de Datos II

Análisis y Curación - Exploración y Curación de Datos

Dr. José Ramón Iglesias
DSP-ASIC BUILDER GROUP
Director Semillero TRIAC
Ingenieria Electronica
Universidad Popular del Cesar

¿De qué se trata?

Situación problemática

Recolección de datos

Análisis y exploración

- ¿Qué variables están disponibles?
- ¿Qué distribución tienen?
- ¿Cómo se relacionan las distintas variables?

Definición de la tarea

- ¿Qué vamos a intentar predecir?
- ¿Qué variables son relevantes?
- ¿Cómo podemos transformarlas?

Diseño de experimentos

- ¿Qué modelos representan mejor el fenómeno?
- ¿Cómo vamos a entrenar?
- ¿Cómo vamos a medir el éxito?

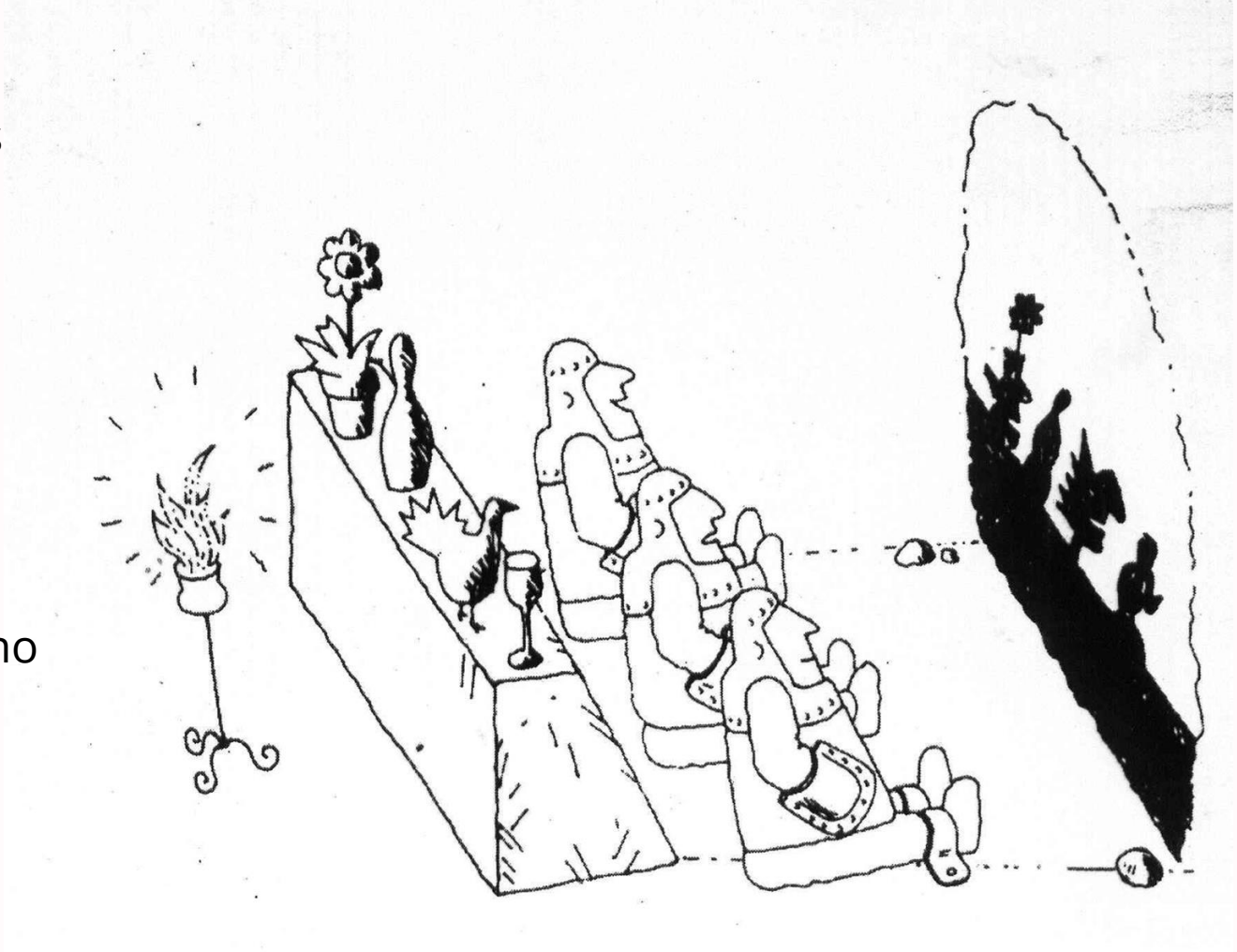
Curación de datos: Seleccionar y transformar los datos para prepararlos para la experimentación

Solución de datos

Queremos hacer emerger las características importantes para una tarea determinada

La ciencia de datos es como la *alegoría de la caverna* de Platón

Los datos son un **proyección** que nos muestra sólo ciertos aspectos del fenómeno que estamos estudiando



Curación de datos

Aspectos conceptuales

- Tratamiento de outliers
- Detección de sesgos
- Imputación de valores

Aspectos prácticos

- Lectura y limpieza
- Agregación y transformación
- Reproducibilidad
- Particionado y sampleo

Exploración de datos

- Para decidir los procesos de curación, tenemos que entender nuestros datos **en conjunto**. Incluye:
  - Todas las herramientas de análisis que hemos visto.
  - Técnicas más complejas para el análisis de datos que permiten relacionar múltiples variables.
  - Técnicas de visualización de datos no estructurados

Situación problemática	Datos	Decisiones de curación
Predecir los salarios de los programadores en Argentina en 2020	Encuesta voluntaria con columnas edad, género, años de experiencia y salario	• Eliminar edades menores que 18 y mayores que 99 • Eliminar salarios mayores que 1 millón de pesos • Estandarizar los años de experiencia de tal forma que la media sea 0. • Re-escalar las edades en un rango de 1 a 0, tal que 18 años o menos corresponda a 0 y 70 años o más corresponda a 1. • Eliminar la columna género.

Situación problemática	Datos	Decisiones de curación
Predecir los salarios de los programadores en Argentina en 2020	Encuesta voluntaria con columnas edad, género, años de experiencia y salario	• Eliminar edades menores que 18 y mayores que 99 • Eliminar salarios mayores que 1 millón de pesos • Estandarizar los años de experiencia de tal forma que la media sea 0. • Re-escalar las edades en un rango de 1 a 0, tal que 18 años o menos corresponda a 0 y 70 años o más corresponda a 1. • Eliminar la columna género.
Predecir el precio de una propiedad	Base de datos gubernamental con registros de transacciones inmobiliarias. Tiene precio, fecha y ubicación.	• Eliminar día y mes de la transacción. • *Escrapear* sitios de compra/venta para extraer información adicional sobre cada propiedad. • Imputar los valores faltantes utilizando estimaciones en base a ejemplos parecidos.

Tradeoff: usar el conocimiento de dominio vs limitar demasiado nuestro modelado

La maldición de las categorías

¿Qué **información** me aporta la dirección de una propiedad?

La dirección de una propiedad en venta es una variable categórica que no puede utilizarse sin transformarla. Intuitivamente, nosotros inferimos la vecindad de una propiedad en base a su dirección, y en base a eso estimamos el valor.

- Las categorías me aportan información porque agrupan distintos ejemplos. Mientras menos ejemplos agrupen, menos informativas son.

La maldición de las categorías

- Eliminar la variable.
- Combinarla con otra variable.
  - Ej: Sólo usamos el zipcode para barrios que tengan más de un código postal, o para diferenciar localidades homónimas.
- Crear nuevas categorías:
  - Agrupar categorías similares.
  - Crear una categoría “otro” para categorías que no tienen muchos ejemplos.

Demo notebook

CIED2 Exploracion.ipynb

Tipos de datos

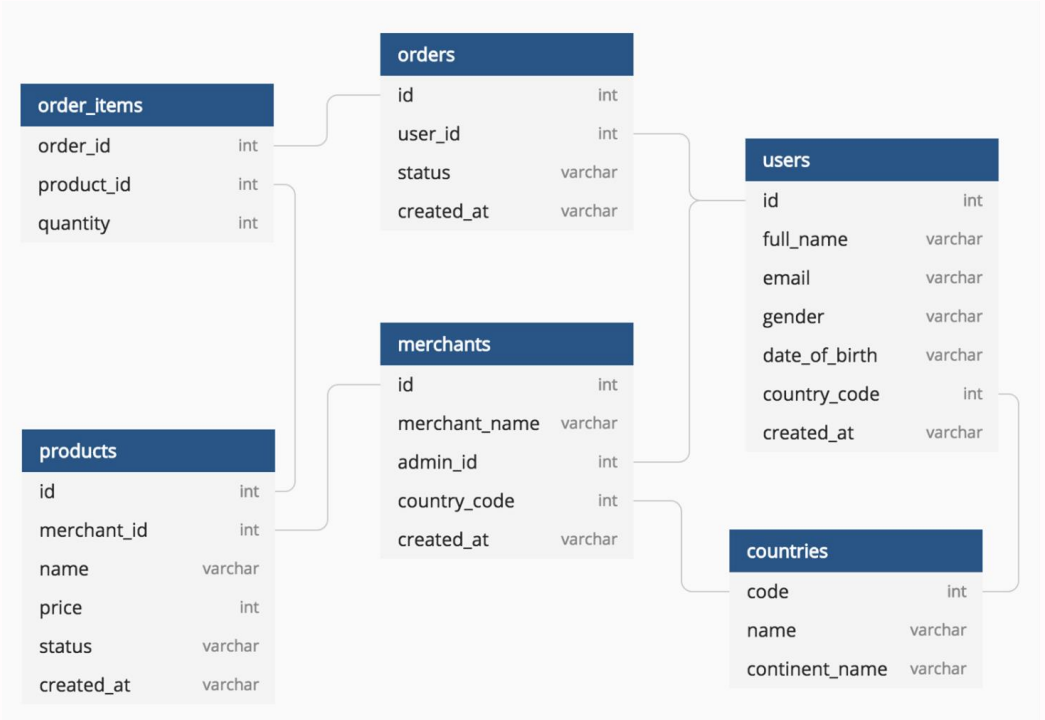
Sí... otra vez...

Estructura de los datos

- Llamamos “datos” a un conjuntos de registros.
- Cada registro tiene asociado un conjunto de características, y las características pueden relacionarse de manera compleja.
- Distintas estructuras suelen almacenarse con formatos de archivo particulares
    - La estructura de los datos no es lo mismo que el tipo de base de datos o archivos en los que se almacena

Datos estructurados

- Todos los registro tienen las mismas características con el mismo tipo
- Las características de algunos registros pueden ser registros en otra tabla



- Archivos en formato CSV, parquet, etc.
- Bases de datos relacionales como MySQL, Postgres

https://www.holistics.io/blog/top-5-free-database-diagram-design-tools/

Datos semi-estructurados

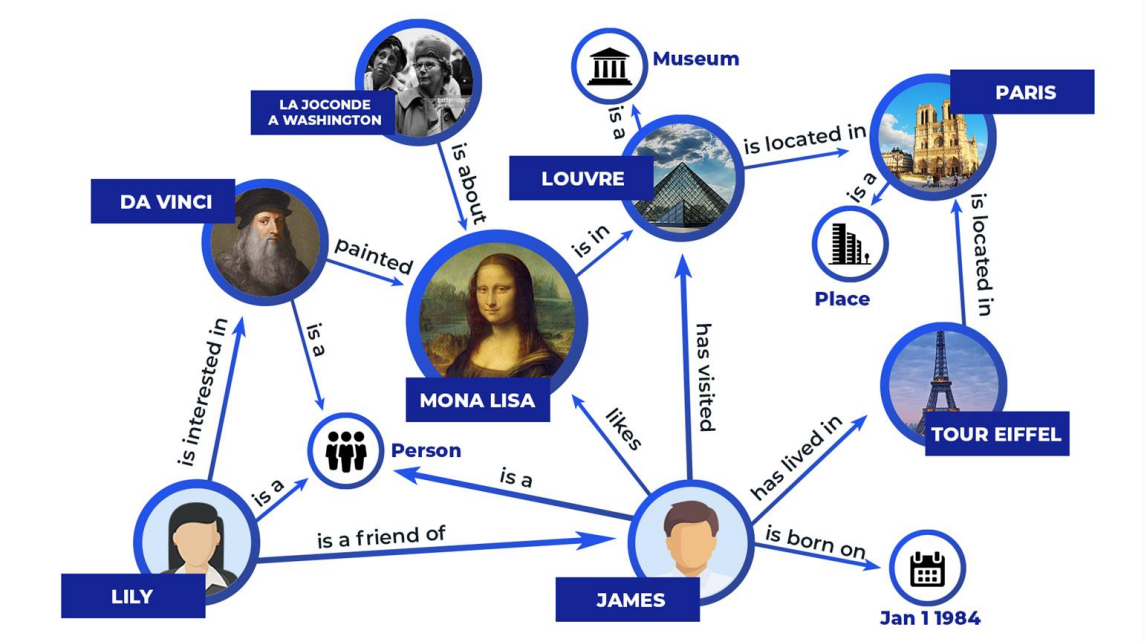
- Cada registro tiene un conjunto distinto de características
- Los registros pueden estar anidados


```
{"orders": [
  {
    "client_id": 1458,
    "items": [
     {"description": "Empanadas", "amount": 12},
     {"description": "Salsa picante", "amount": 1}
    ],
    "total": 950,
    "payment_method": "cash"
  },
  {
    "client_id": 985,
    "items": [
     {"description": "Lomito Completo", "amount": 2,
      "observations": "Uno sin huevo"}
    ],
    "total": 1400,
    "payment_method": "debit",
    "debit_card": "Maestro"
  }
]}
```


- Archivos en formato JSON
- Bases de datos no relacionales como MongoDB

Datos semi-estructurados

- Los registros pueden tener relaciones complejas
  - Jerarquías
  - Estructura de grafo (Twitter)



- Triplas RDF
- Graph-oriented databases

https://www.komunicando.es/wp-content/uploads/2017/12/knowledge-graph.jpg

Datos no estructurados

- Colecciones de distintos tipos:
  - Documentos de texto
  - Imágenes
  - Audio
- Pueden o no tener metadatos asociados

Enriquecimiento de datos

Combinando distintos datasets

Agrupamiento y agregación

1. groupby:
   - Tomar una serie de columnas A, B, C
   - Por cada combinación de valores de las columnas (a1, b1, c1), agrupar las filas que tengan esos valores.
2. agg:
   - Toma una función f
   - Por cada grupo de filas, aplicar la función f a cada columna.

Agrupamiento y agregación

df.groupby('species').agg('sum')

	species	sepal_length	sepal_width	petal_length	petal_width
0	setosa	5.1	3.5	1.4	0.2
1	setosa	4.9	3.0	1.4	0.2
2	setosa	4.7	3.2	1.3	0.2
3	setosa	4.6	3.1	1.5	0.2
4	setosa	5.0	3.6	1.4	0.2
50	versicolor	7.0	3.2	4.7	1.4
51	versicolor	6.4	3.2	4.5	1.5
52	versicolor	6.9	3.1	4.9	1.5
53	versicolor	5.5	2.3	4.0	1.3
54	versicolor	6.5	2.8	4.6	1.5
100	virginica	6.3	3.3	6.0	2.5
101	virginica	5.8	2.7	5.1	1.9
102	virginica	7.1	3.0	5.9	2.1
103	virginica	6.3	2.9	5.6	1.8
104	virginica	6.5	3.0	5.8	2.2

SUM

SUM

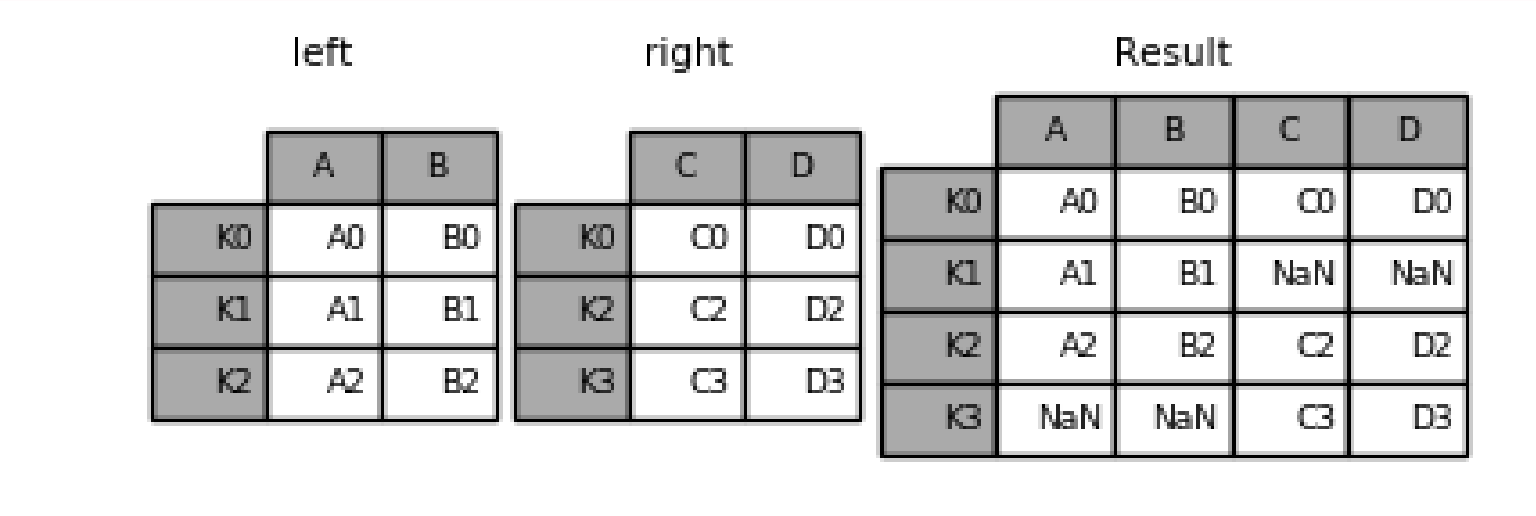
SUM

species	sepal_length	sepal_width	petal_length	petal_width
setosa	24.3	16.4	7.0	1.0
versicolor	32.3	14.6	22.7	7.2
virginica	32.0	14.9	28.4	10.5

https://towardsdatascience.com/how-to-use-the-split-apply-combine-strategy-in-pandas-groupby-29e0eb44b62e

Join y merge

1. df1.join(df2, how='outer')
   - Une horizontalmente los DataFrames y hace coincidir las filas donde el valor del índice es el mismo



	A	B
K0	A0	B0
K1	A1	B1
K2	A2	B2

	C	D
K0	C0	D0
K2	C2	D2
K3	C3	D3

	A	B	C	D
K0	A0	B0	C0	D0
K1	A1	B1	NaN	NaN
K2	A2	B2	C2	D2
K3	NaN	NaN	C3	D3

Join y merge

1. df1.merge(df2, on='key')
   - Igual al join, pero en lugar de comparar los índices, compara un conjunto de columnas.

left

	key	A	B
0	K0	A0	B0
1	K1	A1	B1
2	K2	A2	B2
3	K3	A3	B3

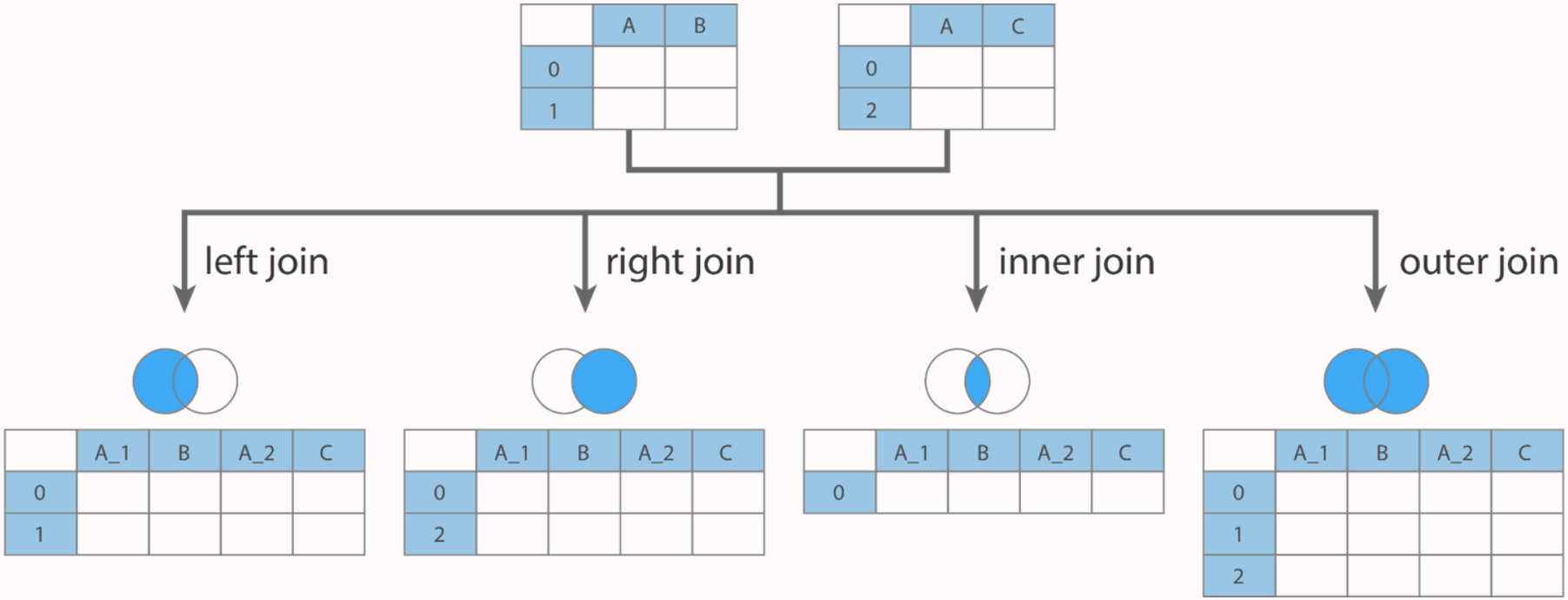
right

	key	C	D
0	K0	C0	D0
1	K1	C1	D1
2	K2	C2	D2
3	K3	C3	D3

Result

	key	A	B	C	D
0	K0	A0	B0	C0	D0
1	K1	A1	B1	C1	D1
2	K2	A2	B2	C2	D2
3	K3	A3	B3	C3	D3

Join y merge



https://towardsdatascience.com/python-pandas-dataframe-join-merge-and-concatenate-84985c29ef78

¡Duplicados imprevistos!

df1

Producto	Ventas
R22	45
J14	10
R5	58
P17	24

df2

Producto	Categoría
R22	Remera
J14	Jean
J14	Pantalón
R5	Remera
P17	Pantalón

```
all_sales = df1.merge(
    df2, on='Producto')
```

Producto	Categoría	Ventas
R22	Remera	45
J14	Jean	10
J14	Pantalón	10
R5	Remera	58
P17	Pantalón	24

```
cat_sales = all_sales\
.groupby('Categoria').sum()
```

Categoría	Ventas
Remera	103
Jean	10
Pantalón	34

```
total_sales =
    all_sales.Ventas.sum()
```

Demo notebook

CIED2 Combinacion de datasets.ipynb