



Analyse et manipulation des données

DigitalLab@LaPlataforme_





What is it about?



Problematic
situation

data collection

**Analysis and
exploration**

- What random variables are available?
- What distribution do they have?

Task Definition

- What are we going to predict/explain?
- What variables are relevant?
- How can we encode them?

Design of
experiments

- Which models best represent the phenomenon?
- How are we going to train?
- How are we going to measure success?

data solution

Problematic
situation

data collection

Analysis and exploration

- What random variables are available?
- What distribution do they have?

Task Definition

- What are we going to try to predict?
- What variables are relevant?
- How can we encode them?

Data curation:
Select and transform
data to prepare it for
experimentation

Design of
experiments

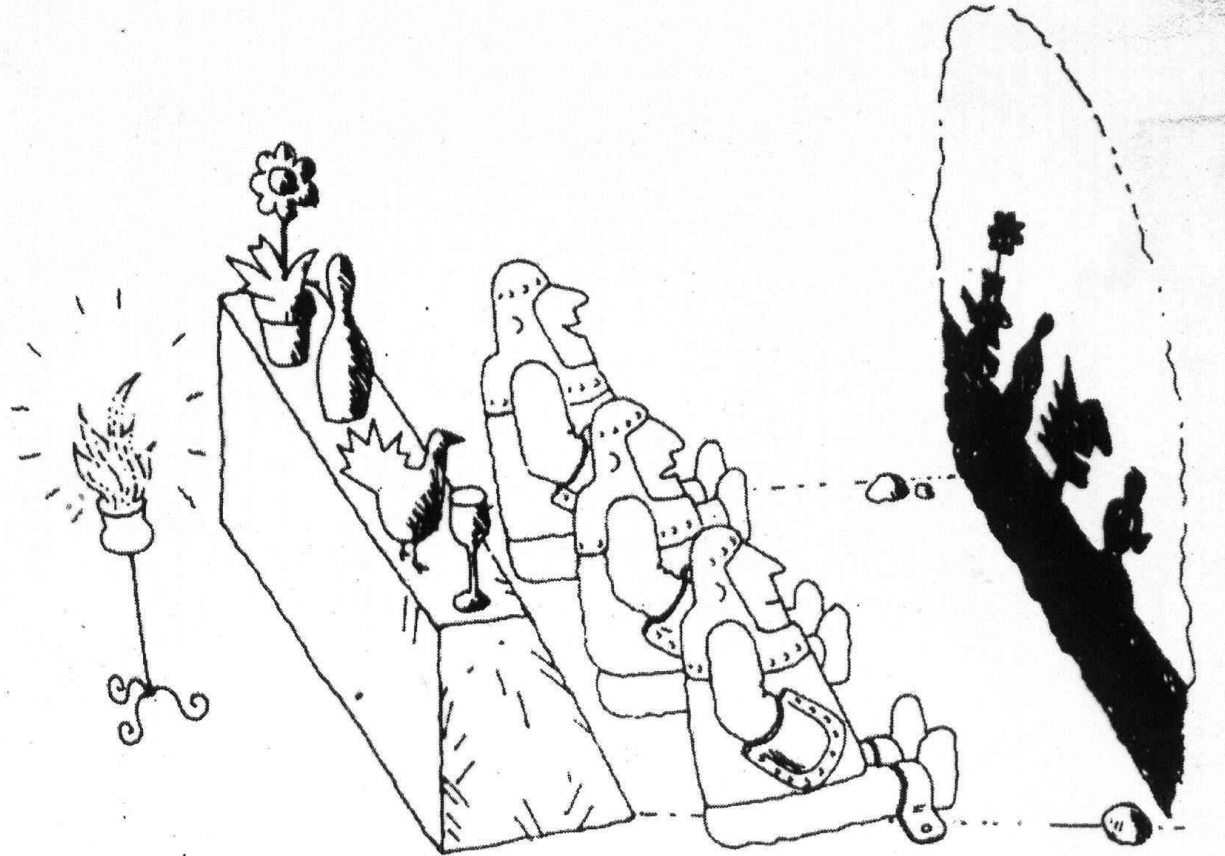
- Which models best represent the phenomenon?
- How are we going to train?
- How are we going to measure success?

data solution

We want to bring out the
important features for a given
task

Data science is like
Plato's cave allegory

The data is a **projection**
that shows us only
certain aspects of the
phenomenon we are
studying.



Data Curation

Conceptual aspects

- Outlier Treatment
- Bias Detection
- Value Imputation

Practical Aspects

- Reading and Cleaning
- Aggregation and Transformation
- Reproducibility
- Partitioning and Sampling

Data Exploration

- To decide on curation processes, we have to understand our data as a whole. It includes:
 - All the analytics tools we've seen.
 - More complex techniques for data analysis that allow multiple variables to be related.
 - Unstructured data visualization techniques

Problematic situation	Data	Curation decisions
Predict programmers salaries in Argentina in 2020	Voluntary survey with age, gender, years of experience and salary columns	<ul style="list-style-type: none"> • Delete ages less than 18 and greater than 99 • Eliminate salaries greater than 1 million pesos • Standardize the years of experience so that the mean is 0. • Rescale the ages in a range from 1 to 0, such that 18 years or less corresponds to 0 and 70 years or more corresponds to 1. • Delete the gender column.

Problematic situation	Data	Curation decisions
Predict programmers salaries in Argentina in 2020	Voluntary survey with age, gender, years of experience and salary columns	<ul style="list-style-type: none"> • Delete ages less than 18 and greater than 99 • Eliminate salaries greater than 1 million pesos • Standardize the years of experience so that the mean is 0. • Rescale the ages in a range from 1 to 0, such that 18 years or less corresponds to 0 and 70 years or more corresponds to 1. • Delete the gender column.
Predict the price of a property	Government database with records of real estate transactions. It has price, date and location.	<ul style="list-style-type: none"> • Delete day and month of the transaction. • <i>Scrape</i> buying/selling sites to extract additional information about each property. • Impute missing values using estimates based on similar examples.

Tradeoff: using domain
knowledge vs limiting our
modeling too much

The curse of the categories

What **information** does the address of a property give me?

The address of a property for sale is a categorical variable that cannot be used without transforming it. Intuitively, we infer the neighborhood of a property based on its address, and based on that we estimate the value.

- The categories give me information because they group different examples. The fewer examples they group together, the less informative they are.

The curse of the categories

- Delete the variable.
- Combine it with another variable.
 - Ex: We only use the zipcode for neighborhoods that have more than one postal code, or to differentiate homonymous localities.
- Create new categories:
 - Group similar categories.
 - Create an “other” category for categories that don't have many examples.

Demo notebook

[01_exploration.ipynb](#)



Data Enrichment

Combining different datasets

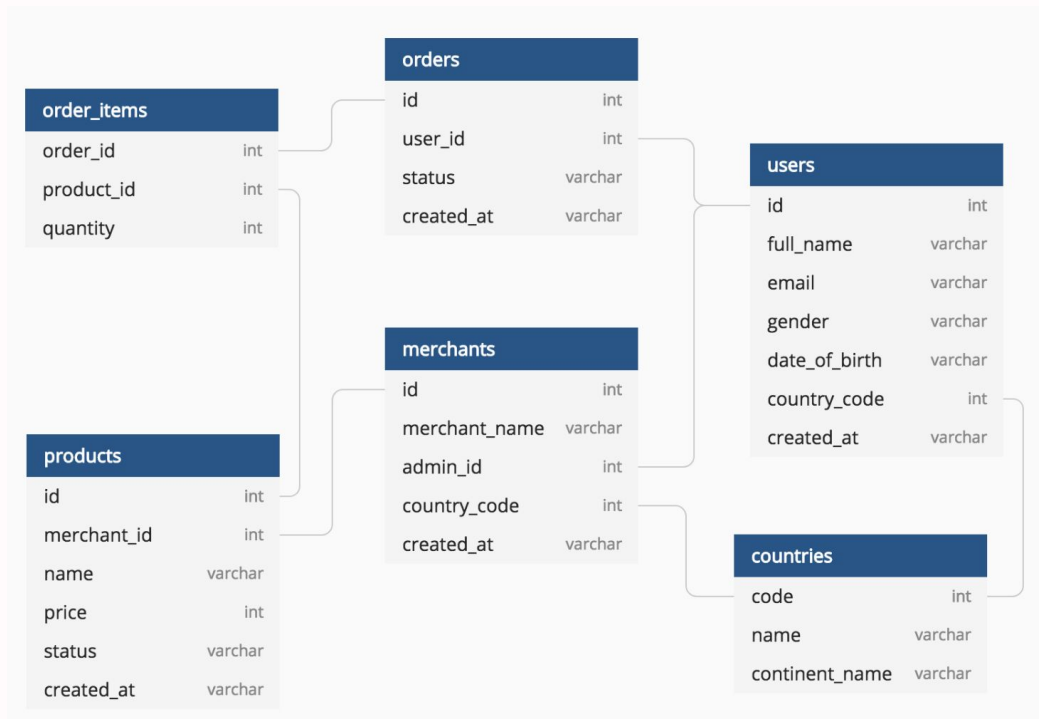


Data Structure

- A dataset is set of records.
- Each record has a set of features associated with it, and the features can be related in complex ways.
- Different structures are often stored with particular file formats
 - The data structure is not the same as the type of database.

Relational Data

- All records in a table have the same characteristics.
- Characteristics of some records may be linked with others in different tables.



- Files in CSV format, parquet, etc.
- Relational databases like MySQL, Postgres

Semi-structured data

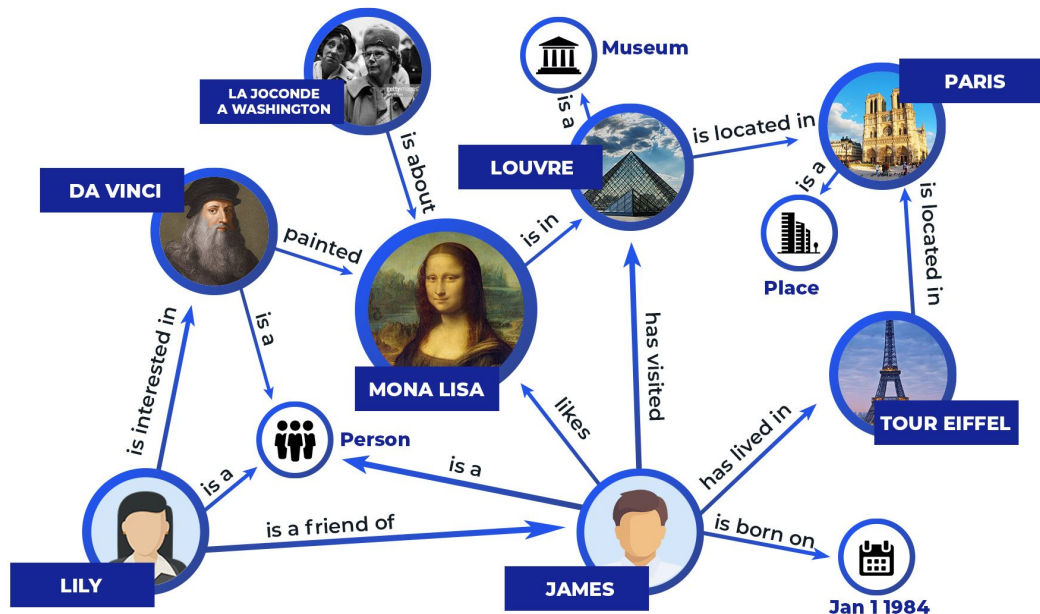
- Each record has a different set of characteristics
- Records can be nested
- One record in a collection doesn't need to have the same features than the others.

```
{  
  "orders": [  
    {  
      "client_id": 1458,  
      "items": [  
        {"description": "Empanadas", "amount": 12},  
        {"description": "Hot sauce", "amount": 1}  
      ],  
      "total": 950,  
      "payment_method": "cash"  
    },  
    {  
      "client_id": 985,  
      "items": [  
        {"description": "Full sandwich", "amount": 2,  
          "observations": "One without egg"}  
      ],  
      "total": 1400,  
      "payment_method": "debit",  
      "debit_card": "Mastercard"  
    }  
  ]  
}
```

- Files in JSON format
- Non-relational databases like MongoDB

Semi-structured data

- Records can have complex relationships
 - Hierarchies
 - Graph Structure (Twitter)



- Triple RDF
- Graph-oriented databases

Unstructured Data

- Collections of different types:
 - Text documents
 - Images
 - Audio
- May or may not have associated metadata



Grouping and aggregation

1. `groupby`:

- Takes a series of columns A, B, C
- For each combination of column values (a1, b1, c1), group the rows that have those values.

2. `agg`:

- Takes a function `f`
- For each group of rows, apply the function `f` to each column.

Grouping and aggregation

`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

	sepal_length	sepal_width	petal_length	petal_width
species				
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

SUM

Join and merge

1. `df1.join(df2, how='outer')`

- Horizontally join the DataFrames and match the rows where the index value is the same

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

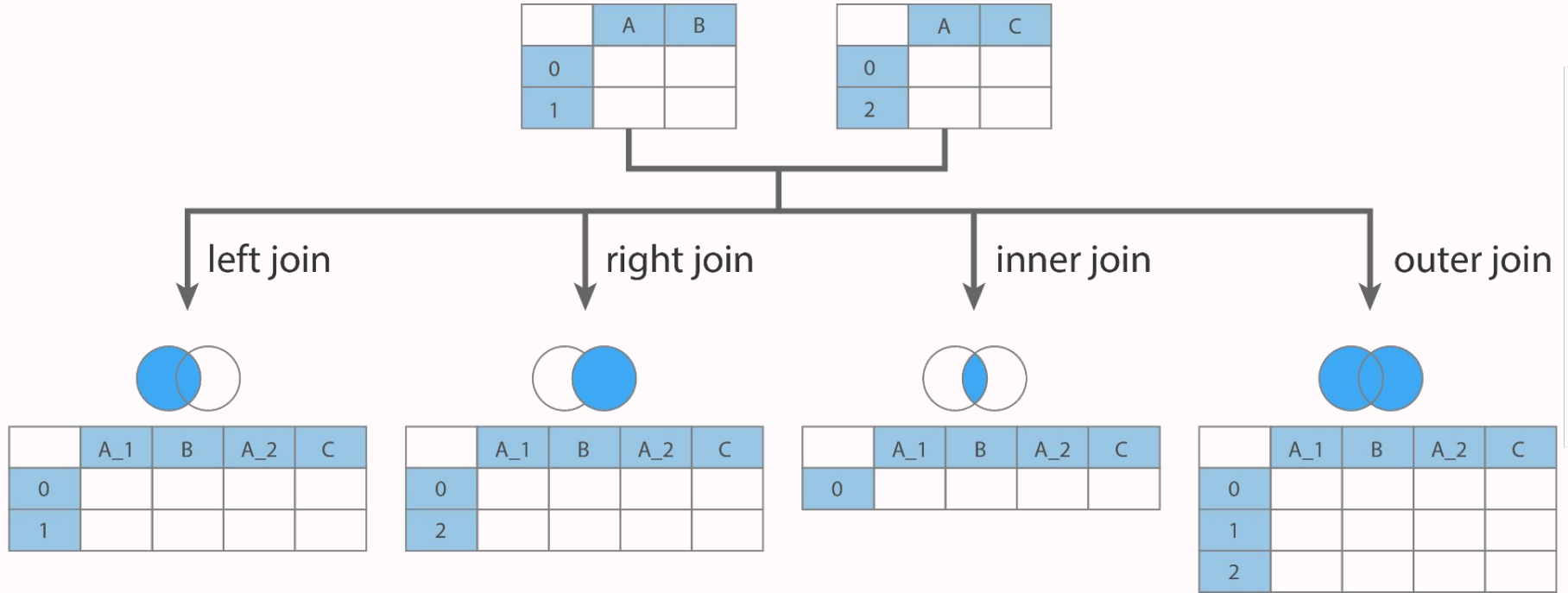
Join and merge

1. `df1.merge(df2, on='key')`

- Same as join, but instead of comparing indexes, it compares a set of columns.

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

Join and merge



Unexpected duplicates!

df1

Product	Sales
R22	45
J14	10
R5	58
P17	24

df2

Product	Category
R22	T-shirt
J14	Jean
J14	Trousers
R5	T-shirt
P17	Trousers

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

Product	Category	Sales
R22	T-shirt	45
J14	Jean	10
J14	Trousers	10
R5	T-shirt	58
P17	Trousers	24

```
cat_sales = all_sales\  
    .groupby(Category).sum()
```

Category	Sales
T-shirt	103
Jean	10
Trousers	34

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



Demo notebook

[02_combining_datasets.ipynb](#)