DataSci 400

lesson 2: nature of data

Seth Mottaghinejad

today's agenda

- **data structures:**
  - tabular, semi-structured, unstructured...
  - trade-offs of each
- **data types:**
  - numeric, categorical...
- **data shapes:**
  - long vs wide

overview of data structures

- different structures or **representation** for data
  - **tabular (structured)**: relational tables (`SQL`), matrices, `DataFrame` , ...
  - **semi-structured**: `JSON` , `XML` , `Mongo DB` , graph datatabases, ...
  - **unstructured**: raw text, images, sound, video, ...
- most ML algorithms only work with tabular data
- once data is made tabular, we still need to do a lot of pre-processing prior to ML

lab time

- here's an example of a single record in semi-structured data:

```
{
  "author" : ["Walter", "Habib"],
  "title" : "How to drink water",
  "language" : "Eng",
  "year_pulished" : ["2008", "2012"]
}
```

- represent it using a **tabular structure** (there is more than one way)
- propose extra columns that can be **extracted** from the above data

choosing data structures

- in a **data lake**, data is in its raw format which includes unstructured and semi-structured
  - data lakes are ideal for **storage**
- for analytics, raw data from a data lake is processed and curated and stored in a **data warehouse** often in tabular form
  - data warehouses are ideal for **analytics**
- **data admins** handle data storage and governance
- **data scientists** handle data transformation and schema conversion

break time

everyone has their jargon

- for ML data almost always needs to be **converted to tabular**
- **rows** - observation, example, record, data points, item, instance
- **columns** - variables, attributes, properties, features, fields, dimensions
  - **target** - label, response variables, dependent variable, outcome
  - **features** - explanatory variable, independent variable, predictors, covariates
    - **numeric**: dates, counts, amounts, etc.
    - **categorical**: grouping variables, identifiers

data types (aka schemas)

- **numeric data**:
  - this is data that we often do **math operations** with
  - **floats** (dollar spend, ratio, percentage, etc.) **integers** (counts, rounded numbers, etc.), **dates and times**
- **categorical data**:
  - also called **grouping variables** because we often group by them when we summarize or visualize the data
  - **low-cardinality** (few groups) vs. **high-cardinality** (many groups)
  - **interactions**: combining multiple variables into one

lab time

- let's return to our example from before

```
{
  "author" : ["Walter", "Habib"],
  "title" : "How to drink water",
  "language" : "Eng",
  "year_pulished" : ["2008", "2012"]
}
```

- after representing it in **tabular structure**, define a schema for each column and justify your choice

break time

lab time

- in the next page, you will see two tabular representation of the semi-structured data below, in **wide format** and **long format**

```
{
  "author" : ["Walter", "Habib"],
  "title" : "How to drink water",
  "language" : "Eng",
  "year_pulished" : ["2008", "2012"]
}
```

- which format do you prefer? why? (think about what happens as more and more books are added to each table)

book title	lang	auth_1	auth_2	pub_1	pub_2
How to drink water	Eng	Walter	Habib	2008	2012
⋮	⋮	⋮	⋮	⋮	⋮

book title	lang	author	published
How to drink water	Eng	Walter	2008
How to drink water	Eng	Walter	2012
How to drink water	Eng	Habib	2008
How to drink water	Eng	Habib	2012
⋮	⋮	⋮	⋮

data shapes

- tabular data can be in **long** format or **wide** format
- if data is very **sparse**, the long format is usually better because it requires less storage
- for data exploration, there is no hard rules about which format to choose, it all **depends on the analysis** and to some extent on **personal preference**
- for machine learning, data **usually** needs to be in wide format
- transforming data from long to wide or vice versa is usually called **pivoting** or **reshaping**

lab time

- in the previous lab, we saw the same data represented in the tabular format in two ways: long and wide
- using by far the most common language for querying tables, `SQL`, write queries against the **long table** and **wide table** to find
  1. the number of authors per book title
  2. the average number of years between the first and second publication of a book
- in each case, which query seem more natural?

further reading

the `pandas` and `seaborn` documentation pages linked below contain good tutorials and lots of examples to learn from

- check out the tutorials on [`pandas`]:
  - https://pandas.pydata.org/
- check out the tutorials on [`seaborn`]:
  - https://seaborn.pydata.org/

the end