Data Science for Urban Mobility

LTAT.06.014
Lecture 2: Data Preparation
Part 1:
“Database Design”
Reflection

Before “Relational Model”

CSV
Reflection

Before “Relational Model”

Directory file model
Reflection

Example 2.1:
Based on our previous python practice, let's assume we have data (csv files) about accidents in different countries and for different years.

Folders:
/accidents/UK/2018
/accidents/EE/2018
/accidents/GR/2018
/accidents/FR/2018

```python
for countryID in os.listdir("accidents/"):  
    for yearID in os.listdir("accidents/"+countryID+"/"):  
        for line in open("accidents/"+countryID+"/"+yearID):  
            print(line)
```
Reflection

Before “Relational Model”

Directory file model

Example 2.1:

Based on our previous python practice, Let’s assume we have data (csv files) about accidents in different countries and for different years.

Folders:

- /accidents/UK/2018
- /accidents/EE/2018
- /accidents/GR/2018
- /accidents/FR/2018

Imagine we are interested in iterating over just accidents from one country or from one year?
Reflection

Before “Relational Model”

Directory file model

Example 2.1:

Based on our previous python practice, let’s assume we have data (csv files) about accidents in different countries and for different years.

Folders:

/accidents/UK/2018
/accidents/EE/2018
/accidents/GR/2018
/accidents/FR/2018

```python
for countryID in os.listdir("accidents/"):  
    for yearID in os.listdir("accidents/"+countryID+"/"):  
        if countryID=="EE":  
            for line in open("accidents/"+countryID+"/"+yearID): 
                print(line)
```
Reflection

Example 2.1:
Based on our previous python practice, Let’s assume we have data (csv files) about accidents in different countries and for different years.

Folders:
- /accidents/UK/2018
- /accidents/EE/2018
- /accidents/GR/2018
- /accidents/FR/2018

```python
for countryID in os.listdir("accidents/"):
    for yearID in os.listdir("accidents/"+countryID+"/"):
        if countryID=="EE":
            for line in open("accidents/"+countryID+"/"+yearID):
                print(line)
```
Reflection

Before “Relational Model”

Example 2.1:
Based on our previous python practice, Let’s assume we have data (csv files) about accidents in different countries and for different years.

Folders:
- /accidents/UK/2018
- /accidents/EE/2018
- /accidents/GR/2018
- /accidents/FR/2018

countryID="EE"
for yearID in os.listdir("accidents/"+countryID+/":
    if countryID="EE":
        for line in open("accidents/"+countryID+/”+yearID):
            print(line)
Reflection

Before “Relational Model”

Directory file model

Example 2.1:

Based on our previous python practice, let’s assume we have data (csv files) about accidents in different countries and for different years.

Folders:

- /accidents/UK/2018
- /accidents/EE/2018
- /accidents/GR/2018
- /accidents/FR/2018

```python
    countryID="EE"
    for yearID in os.listdir("accidents/"+countryID+"/"):  
        if countryID="EE":
            for line in open("accidents/"+countryID+"/"+yearID):
                print(line)
```
Reflection

Before “Relational Model”
- Directory file model
- Hierarchical model
Reflection

Before “Relational Model”
- Directory file model
- Hierarchical model

“Relational Model”
- Forming "Picture of the world"
- Encapsulation
Reflection

“Relational Model”

Forming "Picture of the world"

Encapsulation

Logic Reasoner

Ontology

Database

Files
Reflection

"Relational Model"

Forming "Picture of the world"

Encapsulation

Definition 2.1: Ontology

In philosophy Ontology is the study of "things"
Reflection

"Relational Model"

Forming "Picture of the world"

Encapsulation

Philosophical Ontology:

Is about questioning the basic categories of existence.

How to define a "thing"?

What makes a "thing" different from another one?

Source: https://existentialcomics.com/philosopher/Judith_Butler
Reflection

“Relational Model”

Forming "Picture of the world"

Encapsulation

Philosophical Ontology:

<table>
<thead>
<tr>
<th>Entity</th>
<th>Position</th>
<th>Colour</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mycar</td>
<td>Narva mnt 18</td>
<td>White</td>
<td>Medium</td>
</tr>
</tbody>
</table>

Source: https://existentialcomics.com/philosopher/Judith_Butler
Reflection

“Relational Model”

Forming "Picture of the world"

Encapsulation

Philosophical Ontology:

<table>
<thead>
<tr>
<th>Cars</th>
<th>Position</th>
<th>Speed</th>
<th>Age</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mycar</td>
<td>Narva mnt 18</td>
<td>220</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Events</th>
<th>Entity</th>
<th>Date</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incident 001</td>
<td>mycar</td>
<td>2018-09-22</td>
<td>11h30</td>
</tr>
</tbody>
</table>

Source: https://existentialcomics.com/philosopher/Judith_Butler
Reflection

"Relational Model"

- Forming "Picture of the world"
- Encapsulation

Data Ontology:
- No Redundancy
- Sense of Reference
- Event Design
- Use Case Modelling
- Data Normalisation
- Structured query Language (SQL)

Ontology

Philosophical Ontology
Data Ontology
Part 1: “Data Preparation and Preprocessing”
Data Preparation and Preprocessing

Architecture

**Definition 2.2:** Data Cleaning or Data Cleansing

It is the process of detecting and correcting corrupted or anomalous records in the data. The process itself refers to performing different tasks such as quality check, data deduplication, data analysis, data standardisation, and data normalisation.

**Definition 2.3:** Feature Engineering

Feature engineering is the process of using domain knowledge to extract feature from raw data and used and as mean to make the ML algorithm performance better.
Data Preparation and Preprocessing

1. Missing Data
   - Ignore
   - Fill in
   - Manually
   - Approximation

2. Noisy Data
   - Clustering
   - ML Algorithm
   - Binning
   - Manual by scripting

3. Inconsistent Data
   - External References
   - Knowledge Engineering Tools
Data Preparation and Preprocessing

GPS Traces

- Noise filtering
  - GPS errors
- Time filter
  - Distance filter
- Stop detection
  - Deduplication
- Stop clustering
- Trajectory compression
Data Preparation and Preprocessing

Noise filtering

• GPS Errors

• Latitude and Longitude pointing to the wrong location.

• Map-matching
Data Preparation and Preprocessing

Noise filtering

- GPS Errors
  - Latitude and Longitude pointing to the wrong location.
- Map-matching
  - Canyon effect
Data Preparation and Preprocessing

Noise filtering

• GPS Errors
  • Latitude and Longitude pointing to the wrong location.
• Map-matching
Data Preparation and Preprocessing

Noise filtering

- Time filter
- Sampling rate: time interval for trip extraction.

Source: Xiaocheng Huang et al., 2019

Data Preparation and Preprocessing

Noise filtering

• Distance filter
Data Preparation and Preprocessing

Stop detection

• Deduplication
Data Preparation and Preprocessing

• Stop clustering
Data Preparation and Preprocessing

• Trajectory compression
Data Preparation and Preprocessing

Filtering

• Summary

Raw Data → After time filter → At the end preprocessing