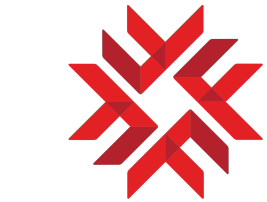
Working with Data Files 4

Course:

INFO-6145 Data Science and Machine Learning



Revised by:
Mohammad Noorchenarboo

September 26, 2024

Contents

Current Section

ETL: Extract-Transform-Load

Definition

ETL is a process used to extract data from various sources, transform it into a usable format, and load it into a system where end-users can access it.

ETL Components

- **Extract:** Gather data from APIs, databases, sensors, etc.
- **Transform:** Clean and convert data into the desired schema or format.
- **Load:** Save the transformed data to a database or target system.

Extract Phase

Sources of Data

Data can be extracted from:

- APIs
- Sensor data
- Databases
- Web scraping (e.g., JSON, XML)

Types of Extraction

- Extract partial data when notified of changes.
- Extract all records updated since the last update.
- Extract full data where the system can't track changes.

Transform Phase

Data Preparation

Once extracted, data needs to be transformed for usability. This involves:

- Data cleaning: removing errors or inconsistencies.
- Data mapping: matching the structure of different datasets.
- Converting data to a format/schema required by the system.

Load Phase

Loading Transformed Data

The final step is to load the transformed data into the target database. The complexity of this depends on:

- Type of database: single server vs. distributed systems (e.g., MapReduce)
- Data volume and variety
- Type of storage: SQL, NoSQL, cloud, etc.

Current Section

Common File Types in Python

You can read various file types from Python, including:

- Plain text files
- Binary files (e.g., images)
- CSV files
- Fixed-width files
- Extensible Markup Language (XML) files

Current Section

Reading Files in Python

Reading Files in Python:

```
# Reading file line-by-line
for line in open('filename.txt'):
    print(line)
```

Reading File in Chunks

```
# Reading file in chunks
with open('filename.txt') as f:
    while True:
        data = f.read(1024)  # Reading 1024 bytes at a time
        if not data:
            break
        print(data)
```

Current Section

Writing CSV Files in Python

Using Pandas

```
df.to_csv('output.csv', sep=',', index=False, encoding='utf-8')
```

Using CSV Module

```
import csv
with open('students.csv', 'w', newline='') as file:
    writer = csv.writer(file)
    writer.writerow(["SNo", "Name", "Subject"])
    writer.writerow([1, "Ash Ketchum", "English"])
```

Current Section

Fixed-Width Files

These were common on mainframe systems where each field took a fixed number of characters.

Example of Fixed-Width Data

```
00001John Doe HR 050000.00
00002Jane Smith IT 106500.01
00003Michael JohnsonFinance 055000.00
```

Current Section

Reading XML in Python

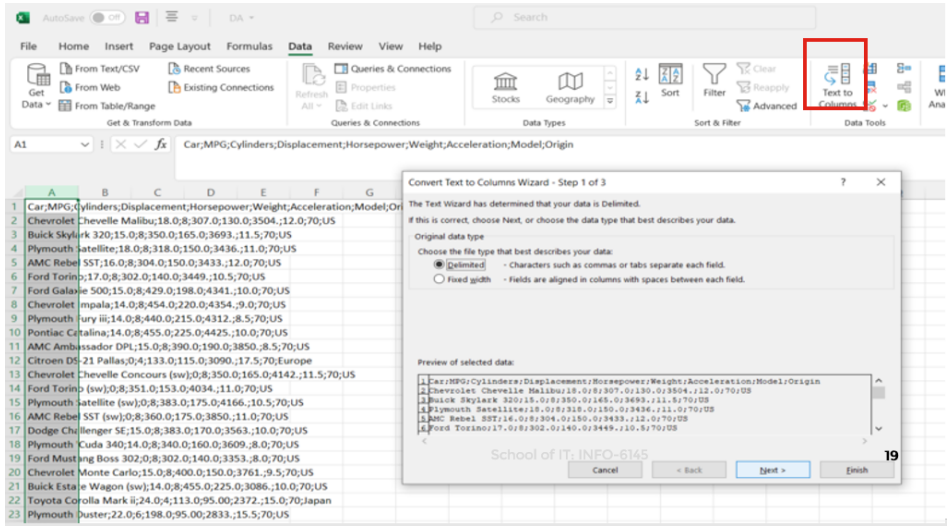
Using BeautifulSoup for XML Parsing

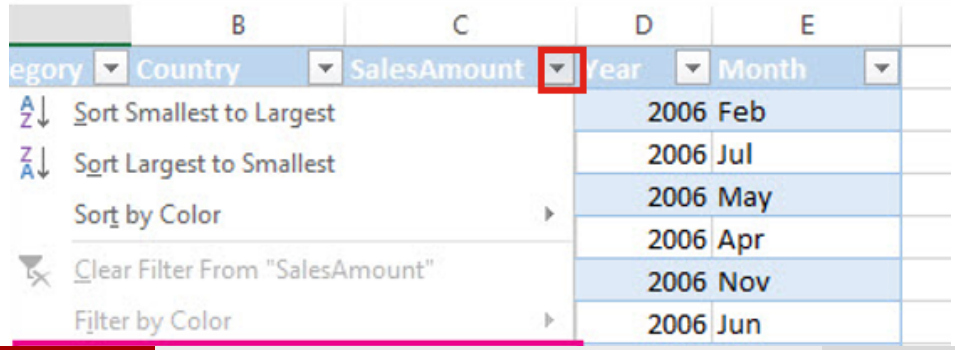
```
from bs4 import BeautifulSoup

with open('file.xml', 'r') as f:
    data = f.read()

Bs_data = BeautifulSoup(data, 'xml')
b_unique = Bs_data.find_all('unique')
print(b_unique)

# Using find() to extract attributes of the first instance
# of the tag
b_name = Bs_data.find('child', {'name':'Frank'})

# Extracting the data stored in a specific attribute of the
#`child` tag
value = b_name.get('test')
```

Current Section

Working with Excel

Using Excel for Data Handling: While Excel is not ideal for large-scale data analysis, it can be useful for smaller datasets, particularly for loading and saving CSV files, or basic filtering and sorting.





Useful Excel Functions

Excel Formulas

Some useful Excel formulas for basic data manipulation:

- `=concatenate(text1, text2, ...)`
- `=len(text)` (string length)
- `=if(condition, true_value, false_value)`
- `=sumifs(sum_range, range1, criteria1, ...)`
- `=vlookup(lookup_value, table_array, column_index_num)`

Current Section

Summary of Data Tools

Topics Covered

- Extract-Transform-Load (ETL)
- Handling Text and Data Files
- Writing Data Files
- Using Excel for basic data manipulation