Massive Data Processing

Master in Computer Science

Jordi Planes

Course 2024

Universitat de Lleida

Introduction

Motivation

This course is designed to provide the knowledge and skills to gather, process and analyze large datasets.

- WebScrapping
- Data Storage Formats and DataFrame Manipulation
- Data Cleaning and Preparation
- Data Visualization

WebScrapping

Web scraping

- Web scraping is the process of extracting data from websites.
- It is a useful technique to collect data from websites that do not provide an API.

WebScrapping

Basic libraries for web scraping in Python are `requests` and `BeautifulSoup`.:

```
1 import requests
2 from bs4 import BeautifulSoup
```

You may also need to use regular expressions to extract specific information from the HTML code.

```
1 import re
```

WebScrapping

Some web pages may block requests from bots. To avoid this, we can set a user agent in the request headers:

```
1 headers = {"User-Agent":"Mozilla/5.0 (Windows...
2 r = requests.get('https://www.abacus.coop/ca/...
3 print('Status code', r.status_code)
```

WebScrapping i

For the titles, we have the following HTML structure:

```
<h3>
<a class="link" href="/ca/....html">Bluey. Libro
    juguete - ...</a>
</h3>
```

That can be parsed with BeautifulSoup:

```
soup = BeautifulSoup(r.text, "html.parser")
# Get the titles
titles = soup.find_all("h3")
```

We have found that there is an texttth3 tag that does not contain a title, so we can filter it out:

WebScrapping ii

```
soup = BeautifulSoup(r.text, "html.parser")
# Get the titles
titles = soup.find_all("h3")
# Filter out the tag without a title
titles = [title.text.strip() for title in titles if
    not title.has_attr('class')]
```

WebScrapping

For the author is easier:

```
<div class="author-brand-wrapper">
<p>Bluey</p>
</div>
```

That can be parsed with BeautifulSoup:

```
# Get the authors
authors = soup.find_all("div",
    class_="author-brand-wrapper")
authors = [author.text.strip() for author in authors]
```

WebScrapping i

For the price is a bit tricky, because there are two prices, the original price and the sales price. The original price is inside a `span` tag with the class `strike-through list` and the sales price is inside a `span` tag with the class `sales` and `value`:

```
<span class="strike-through list ">
<span class="value" content="14.95">
14,95&#8364;
</span>
</span>
...
<span class="sales">
<span class="value " content="14.20">
14,20&#8364;
</span>
```

WebScrapping ii

```
11 </span>
```

That can be parsed with BeautifulSoup, and we can filter out the original price:

```
1 prices = []
2 values = soup.find_all("span", class_="sales")
3 for value in values:
4     price = value.find("span", class_="value")
5     if price:
6     prices.append( price.get("content") )
```

WebScrapping i

We can put all together in a DataFrame:

```
import pandas as pd
df = pd.DataFrame({'Title': titles, 'Author':
    authors, 'Price': prices})
```

And export it to a CSV file:

```
df.to_csv('abacus.csv', index=False)
```

Exercise

- Write a notebook that takes the bestsellers list in WaterSones and returns a DataFrame with the titles, authors, and prices of the books. For example, the title books are in the tag `<div class="title-wrap">`.

Data Storage Formats

Data Storage Formats

- Data storage formats are used to store and exchange data in a structured manner.
- The most common formats include CSV, JSON, XML, and NPZ.

CSV (Comma-Separated Values)

- CSV is a simple file format used to store tabular data.
- Each line in the file represents a row, and the values are separated by commas.
- CSV is widely supported by various applications (e.g. MS-Excel, Apple Numbers, ...) and programming languages.

Example CSV

```
1 title,author,price
2 Bluey. Libro juguete - Donde esta Bluey? (edicion en
      espanol),Bluey,14.20
3 Cor trencat,"Blay, Pep",22.80
4 Tres enigmas para la Organizacion,"Mendoza,
      Eduardo",20.80
5 ...
```

JSON (JavaScript Object Notation)

- JSON is a lightweight data interchange format.
- It is easy for humans to read and write, and easy for machines to parse and generate.
- JSON is commonly used for representing structured data in web applications and APIs.
- JSONL (JSON Lines) is a variant of JSON that stores each JSON object on a separate line.

Example JSON

```
1 {"title":{"0":"Bluey. Libro juguete -
      \u00bfD\u00f3nde est\u00e1 Bluey? (edici\u00f3n
      en espa\u00f1ol)",
2 ...
```

XML (eXtensible Markup Language)

- XML is a markup language that defines a set of rules for encoding documents.
- It is both human-readable and machine-readable.
- XML is widely used for representing structured data in various domains.

Example XML

```
<data>
<row>
    <title>Bluey. Libro juguete - Donde esta Bluey?
        (edicion en espanol)</title>
    <author>Bluey</author>
    <price>14.20</price>
</row>
<row>
<title>Cor trencat</title>
<author>Blay, Pep</author>
```

NPZ (Numpy Zipped Data)

- NPZ is a file format used to store multiple arrays in a single file.
- It is a zipped archive of .npy files, which contain the data and metadata for each array.
- NPZ is commonly used for saving and loading large datasets in scientific computing.
- NPZ files can be read and written using the NumPy library in Python.
- NPZ files are not human-readable, and are best suited for storing large numerical datasets.

Exercises

- Download a CSV file from the internet and load it into a DataFrame, and calculate summary statistics (e.g., `df['column_name'].describe()`).
- Download a JSON file from the internet and load it into a DataFrame, and access nested data structures using keys and indexing (e.g., `data['key1']['key2']`).
- Download an XML file from the internet and load it into a DataFrame, and use `xml.etree.ElementTree.parse()` to parse the file and create an element tree.

Data Exploration

Introduction to Data Processing

Data exploration (EDA) is the process of analyzing data to understand its characteristics, patterns, and relationships.

- data gathering,
- data preprocessing,
- data analysis,
- presentation

Data Cleaning and Preparation

Data cleaning involves identifying and addressing issues such as missing values, inconsistencies, and errors in the data.

- Identifying and handling missing values
- Handling inconsistencies
- Identifying and removing outliers

Hands-on Practice: Data Cleaning and Visualization

In this hands-on session, we will practice data cleaning and visualization techniques using real-world data sets. We will learn to identify and handle missing values, create informative visualizations, and gain insights into the data's characteristics and relationships.

Two of the most common tabular datasets are:

- Iris flower
- Titanic passenger

Two of the most image dataset are:

- MNIST
- CIFAR-10

Data Visualization

```
from sklearn import datasets
import pandas as pd

iris = datasets.load_iris()
iris_df = pd.DataFrame(iris.data)
iris_df.columns = iris.feature_names
iris_df['target'] = iris.target

# If we want to convert integers to floats to make
    data uniform
iris.target.astype(float)
```

Conclusion

- Data exploration provides a foundation for understanding the data, identifying patterns, and preparing the data for further analysis.
- By combining data cleaning, visualization, and statistical techniques, data exploration empowers us to extract meaningful insights and make informed decisions from data.