Python Data Products

Course 1: Basics

Lecture: Processing Structured Data in Python

Learning objectives

In this lecture we will...

- Demonstrate how to read JSON/CSV files into python objects
- Introduce the "gzip" library

Reading data into data structures

- In a previous lecture we saw the basics of how to use the CSV/JSON libraries to read structured data
- What comes next? I.e., how to we read the data into appropriate data structures?

```
In [1]: import csv

In [2]: path = "datasets/amazon/amazon_reviews_us_Gift_Card_v1_00.tsv"

In [3]: f = open(path)

In [4]: reader = csv.reader(f, delimiter = '\t')

In [5]: next(reader)

Out[5]: ['marketplace',
         'customer_id',
         'review_id',
         'product_id',
         'product_parent',
```

Reading data into data structures

- In a previous lecture we saw the basics of how to use the CSV/JSON libraries to read structured data
- What comes next? I.e., how to we read the data into appropriate data structures?

1. How do we read larger csv/json files without having to unzip them?
2. How do we extract relevant parts of the data for performing analysis?
3. What structures make access to the data more convenient?

Code: The gzip library

```
In [1]: import gzip
        path = "datasets/amazon/amazon_reviews_us_Gift_Card_v1_00.tsv.gz"
        f = gzip.open(path, 'rt')

In [2]: import csv
        reader = csv.reader(f, delimiter = '\t')

In [3]: header = next(reader)

In [4]: header

Out[4]: ['marketplace',
         'customer_id',
         'review_id',
         'product_id',
         'product_parent',
         'product_title',
         'product_category',
         'star_rating',
         'helpful_votes',
         'total_votes',
         'vine',
         'verified_purchase',
         'review_headline',
```

"rt" indicates that the file is a text file (default is to read as bytes)

Even this small file is 12mb zipped and 39mb unzipped

Otherwise, the file can be treated like a regular file

- Often we'll want to manipulate files that are cumbersome to fit on disk if we extract them
- The gzip library allows us to read zipped files (.gz) without unzipping them

Code: Reading and filtering files line by line

```
In [5]: dataset = []

In [6]: for line in reader:
            line = line[:-3]
            if line[-1] == 'Y':
                dataset.append(line)

In [7]: dataset[0]

Out[7]: ['US',
         '24371595',
         'R27ZP1F1CD0C3Y',
         'B004LLIL5A',
         '346014806',
         'Amazon eGift Card - Celebrate',
         'Gift Card',
         '5',
         '0',
         '0',
         'N',
         'Y']
```

File is read one line at a time

Drop the text fields

Discard unverified reviews

Two ideas:

1. Read the file one line at a time (rather than reading the whole thing and *then* processing it)
2. Perform filtering as we read the data, so that it is never stored in memory

Code: Reading CSV files into key-value pairs

```
In [5]: dataset = []
```

```
In [6]: for line in reader:
            d = dict(zip(header, line))
            for field in ['helpful_votes', 'star_rating', 'total_votes']:
                d[field] = int(d[field])
            for field in ['verified_purchase', 'vine']:
                if d[field] == 'Y':
                    d[field] = True
                else:
                    d[field] = False
            dataset.append(d)
```

dict(zip(header,line)) makes the line into a **dictionary**

Convert numeric and boolean fields to Python types

```
In [7]: dataset[0]
```

```
Out[7]: {'customer_id': '24371595',
 'helpful_votes': 0,
 'marketplace': 'US',
 'product_category': 'Gift Card',
 'product_id': 'B004LLIL5A',
 'product_parent': '346014806',
 'product_title': 'Amazon eGift Card - Celebrate',
 'review_body': 'Great birthday gift for a young adult.',
 'review_date': '2015-08-31',
 'review_headline': 'Five Stars',
 'review_id': 'R27ZP1F1CD0C3Y',
 'star_rating': 5,
 'total_votes': 0,
 'verified_purchase': True,
 'vine': False}
```

Two ideas:

1. The "dict" operator makes the line into a dictionary, allowing us to index fields by keys (rather than numbers)
2. Convert strings to numbers/booleans where possible

Summary of concepts

- Introduced the gzip library
- Saw some techniques for preprocessing datasets as we read them

On your own...

- Try reading some of the larger Amazon datasets (or the Yelp review data) and compiling statistics from them
- Experiment with the dict() and zip() operators