04-02 Other Forms of Structured Data

Using Pandas DataFrame

In this course, we've talked primarily about structuring data in a tabular form using a DataFrame as our construct. This builds off with spreadsheets which have been used for last 30 years. It's an excellent way to show relationships between instances which are sometimes called observations, entities, or in the case of Pandas, rows, and the attributes that these instances have. Which are often called features or in Pandas, columns.

Advantages of DataFrames

This tabular representation of data is an abstraction that we as data scientists have applied to allow us to more easily build manipulation routines with certain properties. It's an abstraction that allows us to quickly do operations such as summarizing data fields or applying a machine learning algorithm.

Other Representations of Data

Network Diagrams

think one of the most common forms of abstractions that we use is a network diagram. You can think of a network being made up of individuals which have attributes, and that those individuals are connected to other individuals and the connection itself also has attributes. Let's take Twitter data as an example. Individual users have attributes such as their Twitter user ID, their picture, and their name. They can be connected to other individuals.

Tree Structure

More generally though, we can think of networks as being made up of nodes and that these could represent anything, people, sports teams, planets. It all depends on what your data is. The nodes are connected through edges which may be directed or undirected. Sometimes a network is referred to as a graph, and sometimes nodes are referred to as vertices.

An example of trees which you'll learn about is from **natural language processing** or NLP. In NLP it's common to represent a chunk of text as a parse tree, which helps to contextualize ambiguity. Here's an example from the docs of one of the most popular Python libraries for text processing, the Natural Language Toolkit or NLTK. In it they've taken a sentence, the little bear saw a fine fat trout in the brook, and built parse tree based on the English language grammar. You can see that the leaf nodes are the words themselves, which each have one parent node, which is a part of speech tag. For instance, little is an adjective and bear is a noun, and these nodes, each have a common parent. In this case it's been classified as a nominal, which has a parent of a noun phrase. The data scientists can then use this parse tree to find out relationships between words, such as which adjectives refer to specific nouns, an important task in the analytics of product reviews for instance.

Converting Between Data Representations

Being a solid data scientist, means that you're able to take one representation and change it to another to apply techniques you may already know. This is especially important when talking to different clients and collaborators who might have different disciplinary backgrounds.

Adjacency Matrix

For networks, a common second representation is an adjacency matrix. In this case, we might have two matrices, one for following and one for blocking. The rows and columns list all the potential people that we might have linkages between. In the following matrix, we might have a value of true between my row and Paul's column, indicating that I follow Paul. While on the blocking matrix we might have a value of true between Paul's row and Michael. We can just represent these matrices in **NumPy** as we did in the first week of this course.

We can then use libraries such as **NetworkX** in Python to **visualize these networks**, and we can apply certain algorithms to answer interesting questions. Such as, is there an indirect connection between myself, and one of the deans I maybe don't follow like Beth Yakel. This is heavily used in social science research to understand influence and social connections.

Summary

These structures are representations that **we impose** on the underlying data. The **meaning of the data** that we derive from these representations **can change based on how we decide to apply a representation**. A data scientist needs to be able to interact with a broad array of other stakeholders and as such needs to be able to be flexible with how they conceive of and represent data.

In [ ]: