INFO251 – Applied Machine Learning

Lab 3

Satej Soman, Suraj R. Nair

Announcements

- **Problem Set 2 due Feb 11!**

Today

- Any questions from last time?
- Vectorized computation + Matrix handling
- Today's programming tool: `numpy`

Vectorized Computation

- Efficient vectorized computation
- Creating and manipulating matrices in Python
- Matrix operations: Addition, multiplication, dot product

Today's programming tool: `numpy`

How to make a program run fast

- Choice of programming language
  - **Fast:** C, C++, Java, Ocaml, Rust
  - **Slow:** Julia, Python
  - **Very slow:** R

Video: counting to 1 billion in C++ vs Python

- Writing efficient code
  - For-loops vs. vectorized computation – where `numpy` comes in

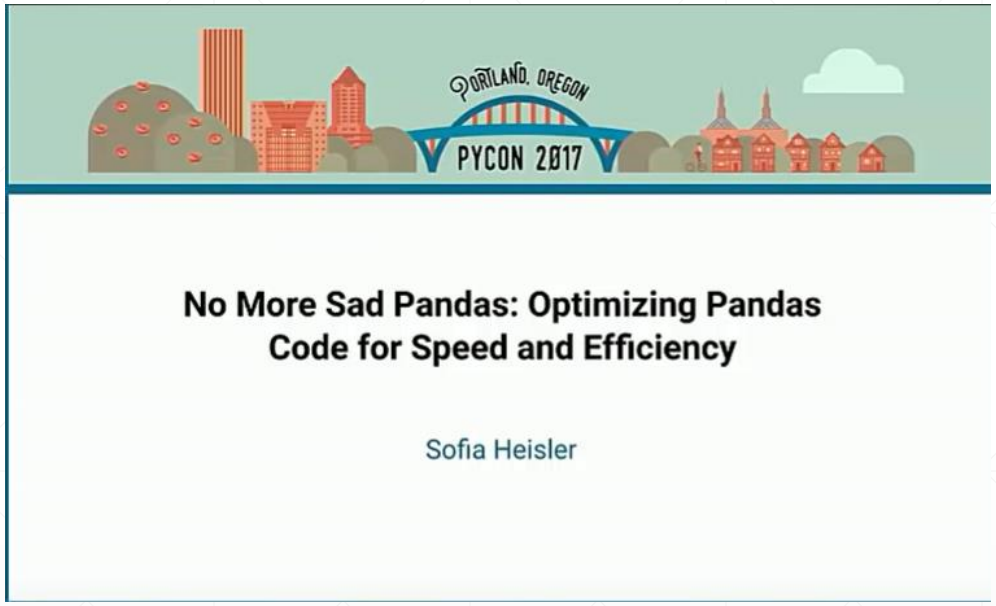
- Hardware and parallelization
  - Run parts of a program in parallel on separate cores -- on a single machine or in a distributed system
  - Libraries for parallelizing/speeding up in python:
    `pyspark, dask, multiprocessing`
  - For more: **CS267**

live coding

Pandas Optimization

- Avoid for loops / `df.iterrows()`
- If looping is a must, use `df.apply(fn)`.
- Pandas series vectorization
- Vector operations on NumPy arrays are more efficient than on native Pandas series
- Consider parallelized/sped-up alternatives:

  `pandarallel, polars, dask`



Video