Scaling Overview

A very common problem in data analysis is standardizing and scaling data

- Many algorithms are sensitive to the range and scale of the inputs
- Text values typically need to be transformed into scalars

The Solution

- Using CSV files as input
- Output CSV files in the same structure

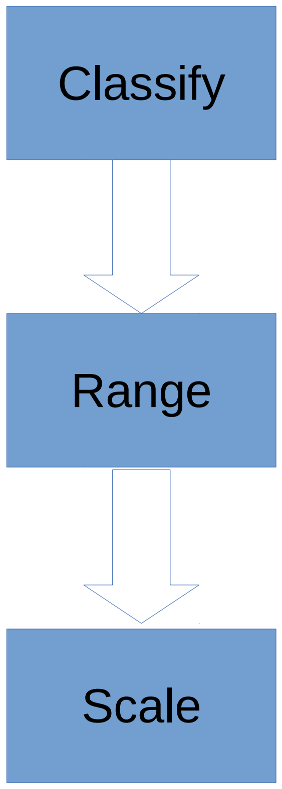
The code is structured so it can run:

- Locally
- On a cloud-based server, ex. Amazon EC2
- As a series of map/reduce jobs, ex. Amazon EMR

Details

- Uses stdin and stdout, which allows working locally as well as conforming to the requirements of Hadoop Streaming
- Built in Python, tested with 2.7.4
-

Basic Flow



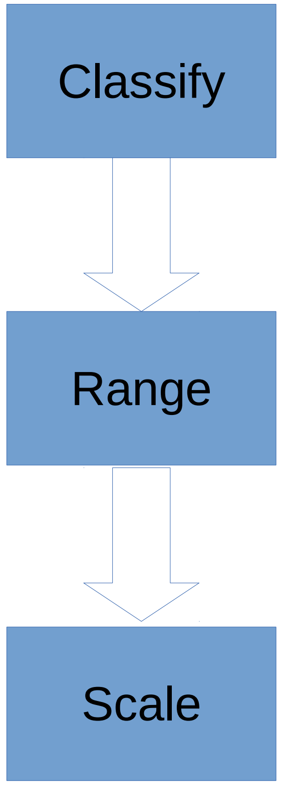
Is the column text or numeric?

If the column is numeric, find the min and max
If the column is text, extract the enumerated list of values

Transform the data

Running Locally

process.sh <file name>
Which wraps the scripts below:



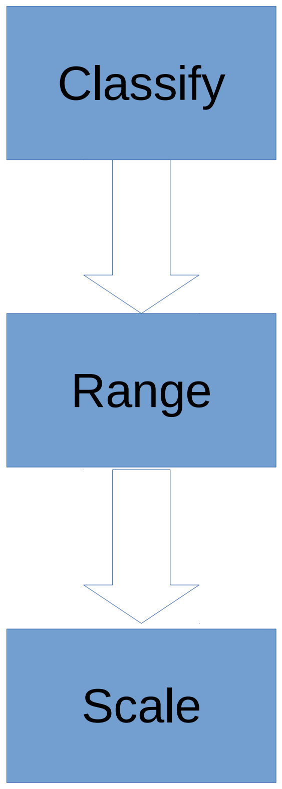
classify.sh <file name>

range.sh <file name>

scale.sh <file name>

As Map/Reduce Jobs

Stage the source file



Set up a job with classify_mapper.py as the mapper and classify_reducer.py as the reducer

Extract the output from the previous step as an argument to this step, and use range_mapper.py and range_reducer.py

Provide the result from the Range step as well as the source file, and use scale_mapper.py as the mapper and scale_reducer.py as the reducer

The resulting output will be a CSV with the columns in the same order and the data transformed