Thanks for posting this @arbitrage! I wasn't aware of Wes McKinney's pet file format feather

before you'd mentioned it. I finally got around trying it today. The load time seems significantly faster than what it takes to load a compressed csv (albeit, lzma

, the format used by xz

achieves much higher compression ratios than lz4

which doesn't compress as well, but is much faster).

I ran the code you'd posted and some more and got these numbers:

file name

size

comments

latest_numerai_training_data.csv.xz

51M

The original xz compressed csv file

latest_numerai_training_data.feather

371M

Feather with no compression

latest_numerai_training_data.feather.lz4

241M

Feather with LZ4 compression

latest_numerai_training_data.parquet

64M

Parquet

Curiously enough, the parquet file is smaller than the compressed feather file. It defaults to using snappy

for compression, which is quite similar to lz4

in its performance characteristics (high speed, low compression). Also, parquet

is an older, more widely used format (ask anyone who's written Spark pipelines) and supports JVM (h/t @bor

) and C++, in addition to the few languages that feather supports.

It's worth noting that the reason why both feather

and parquet

seem to load dataframes instantaneously is because they usemmap, which has better performance, but also the cost of reading the file is amortized over the entire read (i.e although the load call returns instantaneously, the actual data is only loaded lazily from disk, as needed), while read_csv

has to read the entire file into memory at once. There areother ways to make it even faster, but since we're dealing with interpreted languages like Python and R, it's probably not worth the effort.

Here's a code snippet for serializing any Pandas dataframe into a parquet file.

import pyarrow as pa import pyarrow.parquet as pq

def write_parquet(df, filename="data.parquet"): table = pa.Table.from_pandas(df) pq.write_table(table, filename)

And reading parquet files from pandas is as easy as:

df = pd.read_parquet("data.parquet")

This makes me wonder: Is there a good reason for the feather format to even exist?

The whole situation is reminiscent of this XKCD comic.