I/O Optimization in Data Projects

If you use CSVs all the time, read this.

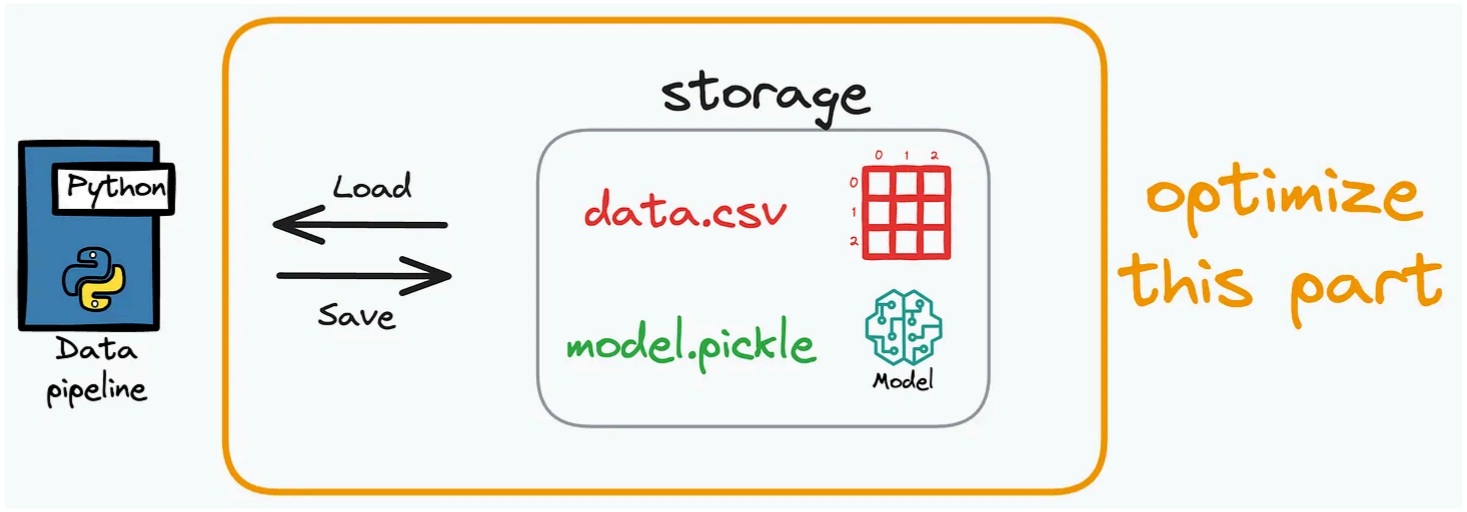
AVI CHAWLA
MAY 31, 2024

34 1 1 Share

Advertise to 77k readers | *Deep Dives*

In a typical DS/ML project, optimization efforts (run-time and memory) are typically dedicated to the algorithmic side.

However, one can also experience profound improvements by optimizing the I/O-related operations.



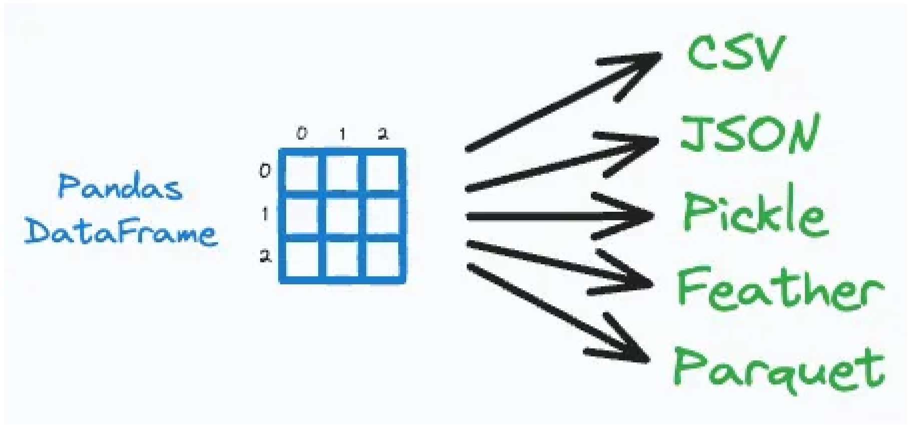
This is because when it comes to data, the most common storage option is a CSV (for small to intermediate projects).

But unknown to many, they are among the **most time-inefficient and storage-inefficient** storage formats you could ever utilize in your data pipeline.

If you use CSVs all the time, then this post is for you.

Let's understand in detail!

In addition to a CSV, there are various other options that we can dump a DataFrame to:



So comparing their performance on the following parameters can be pretty useful to optimize our data pipeline:

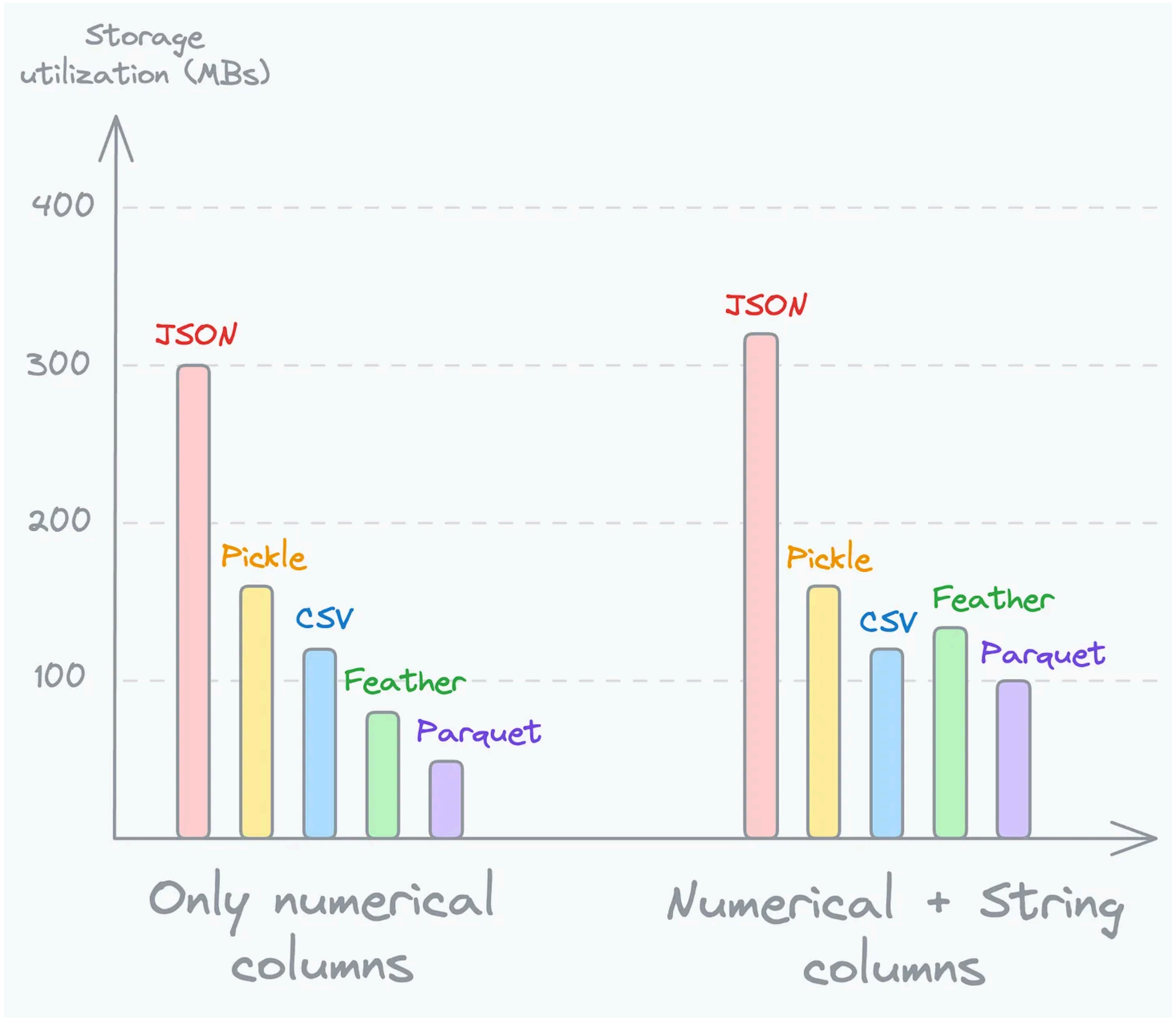
- The space they occupy on disk.
- The read and write run-time.

To conduct this experiment, I created a dummy dataset with:

- 1M rows
- 20 columns:
  - **Case 1:** With only numerical columns.
  - **Case 2:** With 10 numerical and 10 string columns.

> *The study applies to single-node processing.*

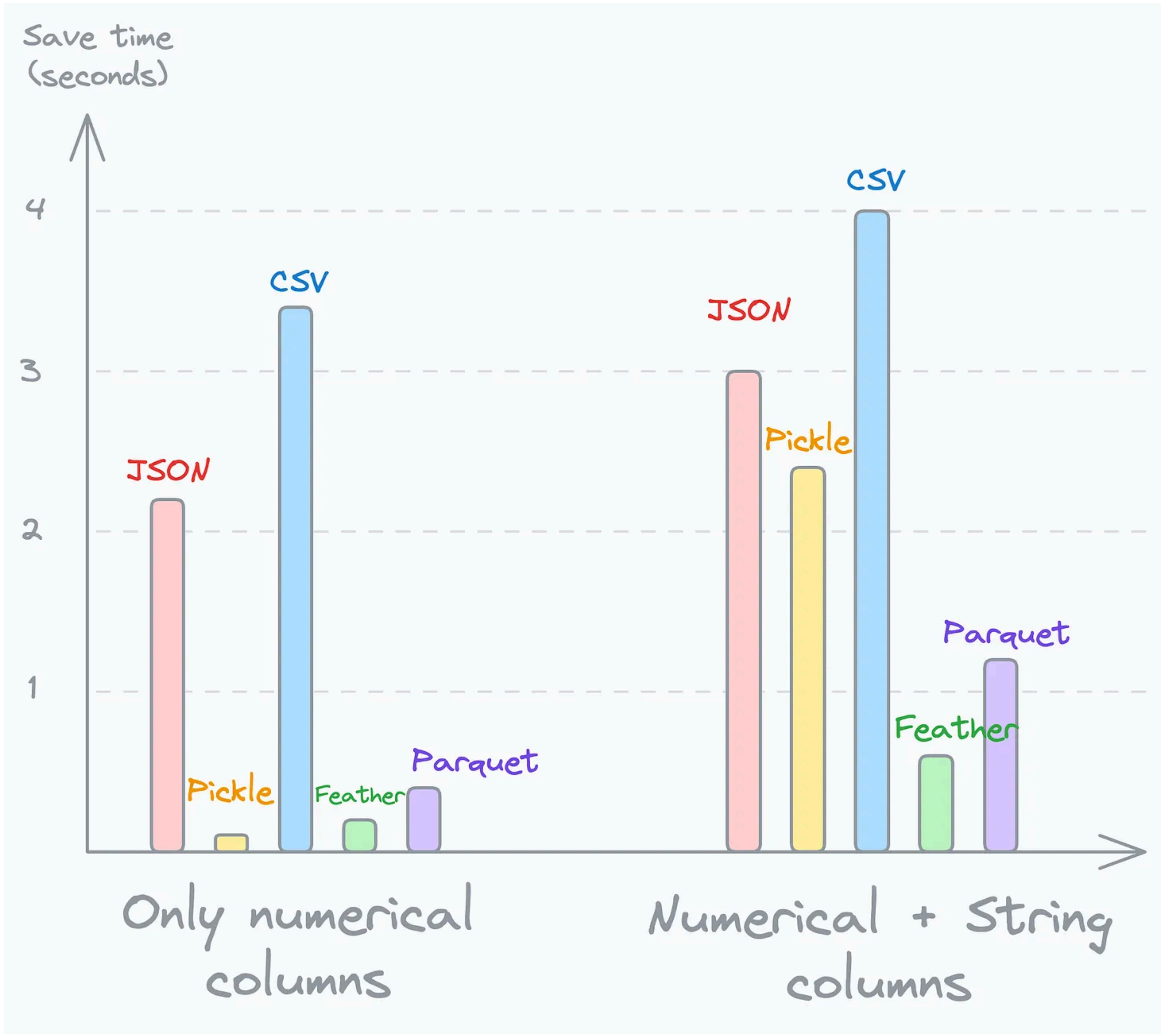
The following visual depicts the memory utilization of these DataFrames (Case 1 and Case 2) when stored in each of the above formats:


Storage utilization (MBs)
400
300
200
100
JSON
Pickle
CSV
Feather
Parquet
Only numerical columns
JSON
Pickle
CSV
Feather
Parquet
Numerical + String columns

As depicted above:

- In both cases, CSVs are not the most optimal format to dump DataFrame to.
- For numerical data, Parquet consumes over 2x less space than CSV.
- For numerical data, Feather is also promising.
- In the case of mixed columns, except JSON, all formats take up almost equal space.

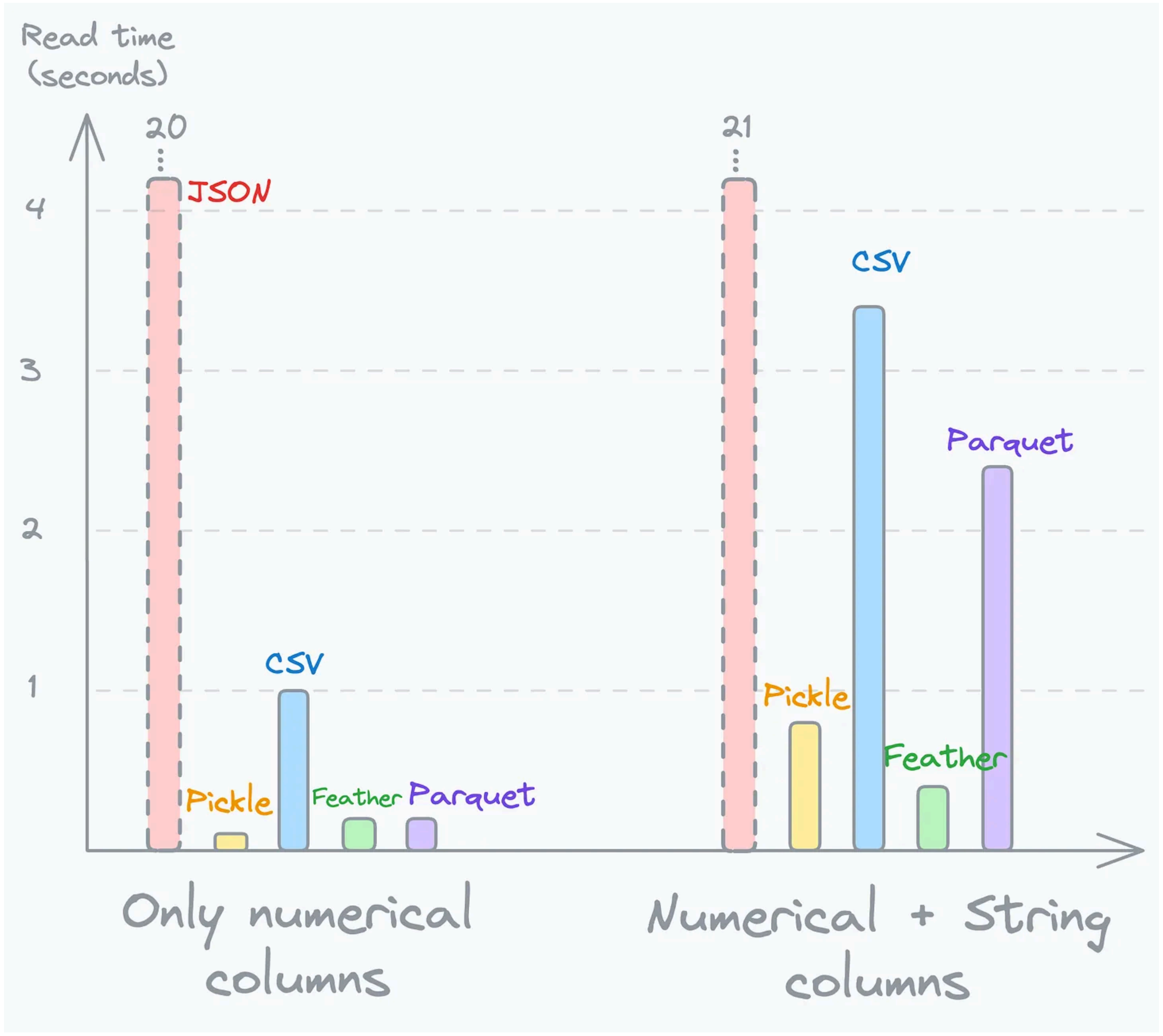
Next, the following visual depicts the run-time to **store** the DataFrame in those formats:


Save time (seconds)
4
3
2
1
JSON
Pickle
CSV
Feather
Parquet
Only numerical columns
JSON
Pickle
CSV
Feather
Parquet
Numerical + String columns

As depicted above:

- Saving a DataFrame to **pickle** is ridiculously faster than CSV when we only have numerical columns. But this run-time drastically increases when we have mixed columns.
- Feather appears to be a clear winner because it has the least saving run-time in both cases. Also, recalling the storage results above, the Feather format was among the most promising formats.
- Here, the Parquet format is also promising.

Finally, the following visual depicts the run-time to **load** the DataFrame from memory when stored in those formats:


Read time (seconds)
20
21
4
3
2
1
JSON
Pickle
CSV
Feather
Parquet
CSV
Parquet
Pickle
Feather
Only numerical columns
Numerical + String columns

As depicted above:

- Read time from Pickle, Feather, and Parquet is the lowest when we have numerical columns. But Parquet's read time drastically increases when we have mixed columns.
- Yet again, Feather appears to be the most promising format, with low run-time in both cases.

Putting it all together

I understand this might be a lot of information, so let me summarize it and provide clear action steps.

1. **CSVs are rarely an optimal choice:**
   a. The only reason you would want to use them is when you want to open the data in tools like Excel.
   b. Another reason is when you want to constantly append new data to the CSV file. Other formats do not support this operation.
   c. In addition to memory and run-time issues, another problem is that CSVs cannot preserve data type information.
      i. For instance, if you change the data type of a column (from, say, `int` to `str`) → store the DataFrame to a CSV → load it back, CSV format will not preserve the data type change.

   ii. After loading the DataFrame, you will get back the `int` data type for that column.

2. Pickle is a promising choice:
   a. **Pickle's storage utilization is unaffected by file format.**
   b. But its read and write run-time suffers in the case of mixed columns.
   c. **Also, remember that the pickle format is Python-specific. You are restricted to Python when you use it.**
   d. That being said, unlike CSVs, Pickle preserves data type information.
3. Comparing Feather and Parquet:
   a. **Feather consumes slightly more memory, but it's the most optimal choice for run-time improvements.**
   b. Parquet is good, but similar to the Pickle format, it is well suited for numerical data only.
   c. **If your priority is to minimize storage utilization, Parquet is better.**
   d. **If your priority is to optimize run-time, Feather is better.**

Of course, please note that these experiments may not entirely translate to every use case.

The idea of this post was to:

- Make you aware of the problems CSV format imposes.

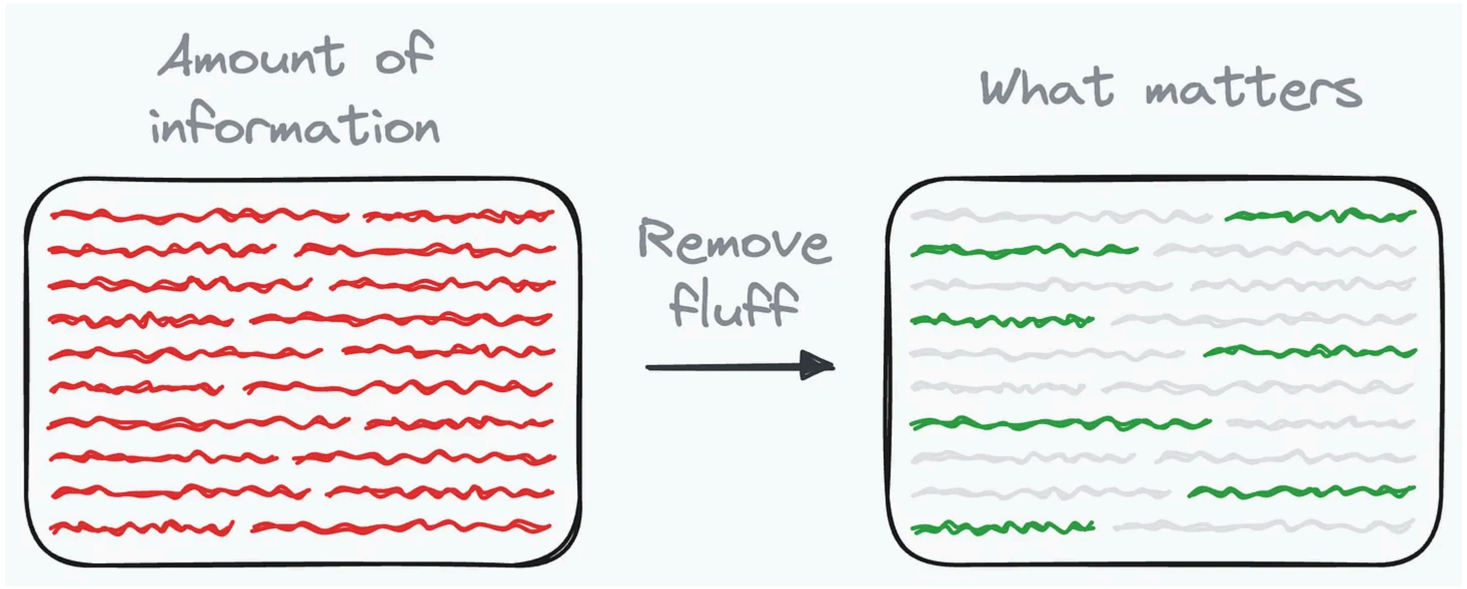
- Share insights on what is typically the most optimal choice and when.

If you want to run this experiment yourself, download this notebook: Storage Format Optimization Experiment.

👉 Over to you: What are some other sources of optimization in data pipelines that many overlook?

Are you overwhelmed with the amount of information in ML/DS?

Every week, I publish no-fluff deep dives on topics that truly matter to your skills for ML/DS roles.



For instance:

- A Beginner-friendly Introduction to Kolmogorov Arnold Networks (KANs).
- 5 Must-Know Ways to Test ML Models in Production (Implementation Included).
- A Beginner-Friendly Guide to Multi-GPU Model Training.
- Understanding LoRA-derived Techniques for Optimal LLM Fine-tuning
- 8 Fatal (Yet Non-obvious) Pitfalls and Cautionary Measures in Data Science
- Implementing Parallelized CUDA Programs From Scratch Using CUDA Programming
- You Are Probably Building Inconsistent Classification Models Without Even Realizing.

- How To (Immensely) Optimize Your Machine Learning Development and Operations with MLflow.
- And many many more.

Join below to unlock all full articles:

SPONSOR US

Get your product in front of 77,000 data scientists and other tech professionals.

Our newsletter puts your products and services directly in front of an audience that matters — thousands of leaders, senior data scientists, machine learning engineers, data analysts, etc., who have influence over significant tech decisions and big purchases.

To ensure your product reaches this influential audience, reserve your space **here** or reply to this email to ensure your product reaches this influential audience.

34 Likes · 1 Restack

Previous Next

1 Comment

Write a comment...

spencer Jun 3

this is all for loading into pandas df's, correct? would be great to get an expanded piece on dbs. context for us is we're debating precompute vs model endpoint for an app, but our model is a gurobi model object

LIKE REPLY SHARE