Building Tecton's Feature Engineering Platform on DuckDB

Mike Eastham
Tecton

- Series C, ~5 years old, ~50 engineers
- We make a Feature Engineering platform which allows customers to define Features for ML models as transformations on top of various data sources. We orchestrate pipelines to compute, store, and serve these values to models.
Problems

Spark is Confusing
Customers respond with confusion when exposed to Spark configuration or debugging

Deployment Complexity
We require a managed Spark vendor (Databricks, EMR, Dataproc) for deployment

Heavyweight
Spark clusters take 10+ minutes to start up, have thousands of configuration parameters. Overkill for many of our customers with datasets in the 10s of GiBs range or less
Requirements

Familiar Devex

- Pandas
- SQL

Easy to Deploy

No *mandatory* third party vendors like we have with Spark

Local First

- Fast to start
- Standalone
- Performant

Integrations

- Snowflake
- BigQuery
- RedShift
- Postgres
- ...

Why DuckDB

Arrow

Seamless integration allows us to flexibly integrate new data sources, transformation modes, etc.

Performance

Fast & Lightweight

Can process non-trivial datasets with modest amounts of RAM due to streaming, out-of-core

Simplicity

Eliminates the complexities of distributed query engines, Spark/Hadoop configuration, etc., which are unnecessary for the datasets we’re typically dealing with.
Results (so far)

- Currently in Private Preview
- Many jobs now complete faster than a Spark cluster can start up
- Scales up to datasets of 10s of GiBs (both input and output) without problems
- Demos and Deployments have been substantially faster
Challenges

- Extension development / distribution
- Managing AWS Credentials
- Delta Lake integration
- pandas / arrow version wrangling
That’s It!

- Find me later if you have questions!
- We’re hiring in SF, NYC, and remote!