Consulting Soft Skills

Chris Shurtleff

August 22, 2022

What is a Data Lake?

Data Lake:

- Like a regular lake, filled with all sorts of interesting garbage.
- If you need something, get a magnet! (High-effort retrieval)

Data Lake vs Data Warehouse

- Data Warehouse has *structured* data
- Data Lake has *unstructured* data
- Structured data requires knowing *precisely* how to describe what you want to keep in advance
- Unstructured lets you keep some data that you *might* want to use later
- Semi-structured data lets you apply *some* structure to otherwise unstructured data

Serverless architecture

What is serverless?

- Using somebody else's computers
- Like sharing these scooters, but for computers!

Serverless Pros

- Lower overall costs

Serverless Pros

- Lower overall costs
- Replaces fixed costs with variable costs

Serverless Pros

- Lower overall costs
- Replaces fixed costs with variable costs
- Complicated compliance and security issues are someone else's problem!

Serverless Pros

- Lower overall costs
- Replaces fixed costs with variable costs
- Complicated compliance and security issues are someone else's problem!
- Extremely easy to ensure continual access to data

Serverless Pros

- Lower overall costs
- Replaces fixed costs with variable costs
- Complicated compliance and security issues are someone else's problem!
- Extremely easy to ensure continual access to data
- Set up is easier

Serverless Pros

- Lower overall costs
- Replaces fixed costs with variable costs
- Complicated compliance and security issues are someone else's problem!
- Extremely easy to ensure continual access to data
- Set up is easier
- Changing things is easier

Serverless Cons

- Relatively easy to accidentally pay for *significantly* more than you need.

Serverless Cons

- Relatively easy to accidentally pay for *significantly* more than you need.
- Somebody else *owns* and *secures* the physical computers

Serverless Cons

- Relatively easy to accidentally pay for *significantly* more than you need.
- Somebody else *owns* and *secures* the physical computers
- Can be more expensive than not serverless alternatives

ETL Pipeline

Modern MLOps

Machine Learning

- Set of algorithms that can be broadly applied across different sets of data
- Mostly applied for *prediction* and *classification*
- Examples:
  - Given sensor data, what is the operator of this machine *doing* (classification)
  - Given the state of a chess game, which move will lead to an ultimate victory (prediction)

How to make an ML model

Generic Supervised approach:

- Build set of *features* from input data
- Determine the *label* of each set of *features*
- *Train* the features, finding the right *weights* that lead from input to label
- *Validate* the weights and algorithm writ large
- *Test* the weights to determine *accuracy* and *discrimination*

Applied ML Models (MLOps)

Automate the last slide!!!

- Automatic data (or ETL) pipelines to generate *features* from input data
- Apply the ML model to the new *features*, creating a set of *labels*
- Use the *labels* as you see fit!
- *IMPORTANT* Validate the algorithm over time, to make sure the relationship doesn't shift!

MLOps in practice

- .yaml, .json, and .py files to define the ETL *infrastructure*
- .yaml, .json, and .py files to define the ETL *pipelines* for feature generation from input data
- .yaml, .json, and .py files to define the compute necessary for finding the correct *weights*
- .yaml, .json, and .py files to define the compute necessary for applying *weights* to new data
- .yaml, .json, and .py files to define where the *labels* and *features*
- Unit tests for checking various steps for issues
- Logs for troubleshooting problems
- Validation: Make sure the relationship between *features* and *labels* doesn't shift over time!