

# Managing Production Data Prep Pipelines

---

**Dr. Venkata Pingali**  
**pingali@scribbledata.io**  
Scribble Data



Anecdotally only **2%** of models\* are **productionized!**

\* Most of these are in Python

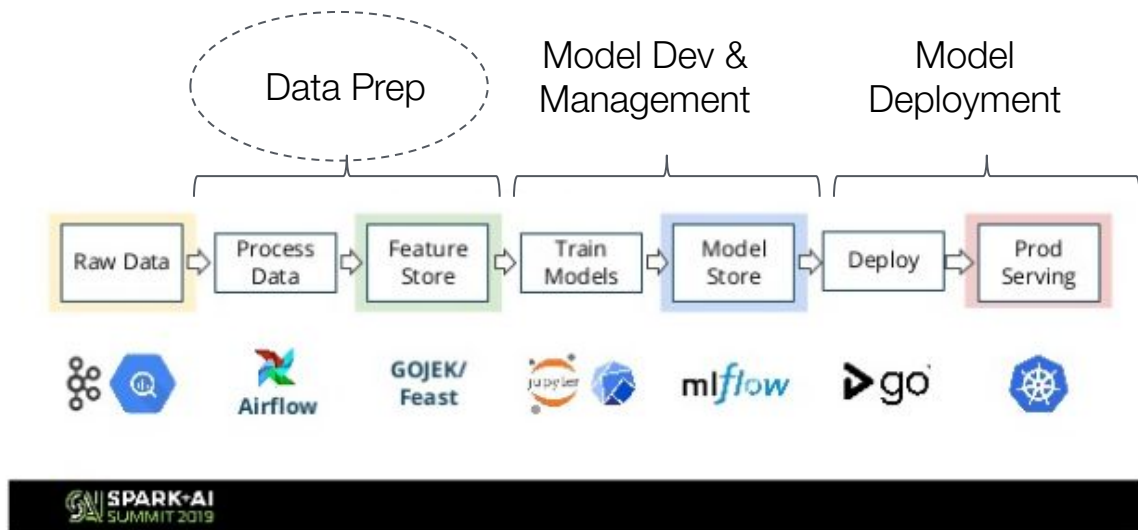
# Outline

---

- ML Infrastructure Overview
  - Why data prep is important
- Pipeline Structure and Challenges
- Where Does Time & Effort Go
- Required Capabilities

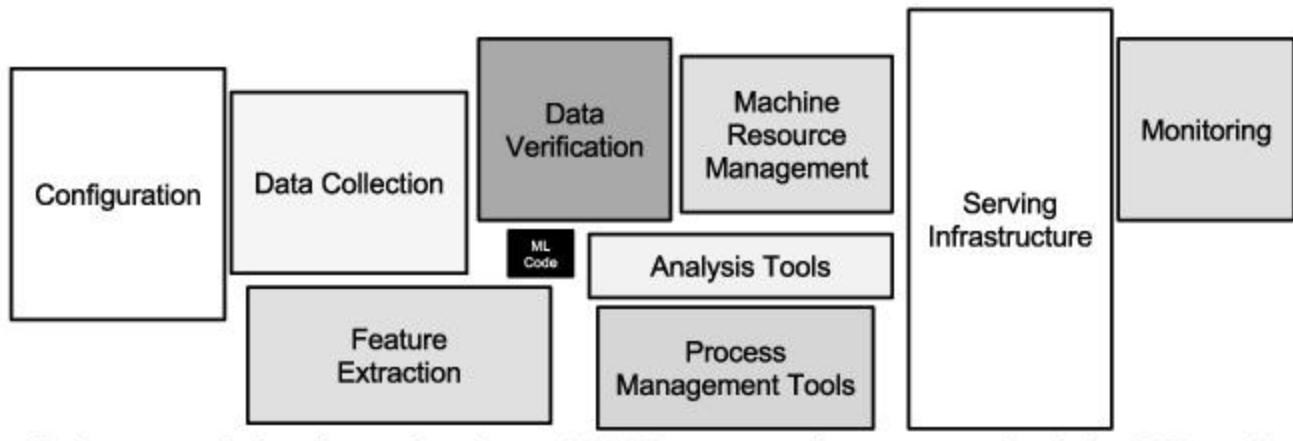
# ML Infrastructure Overview

# Production ML - Emerging Generic Architecture



GoJEK @ Spark AI Summit, April 2019

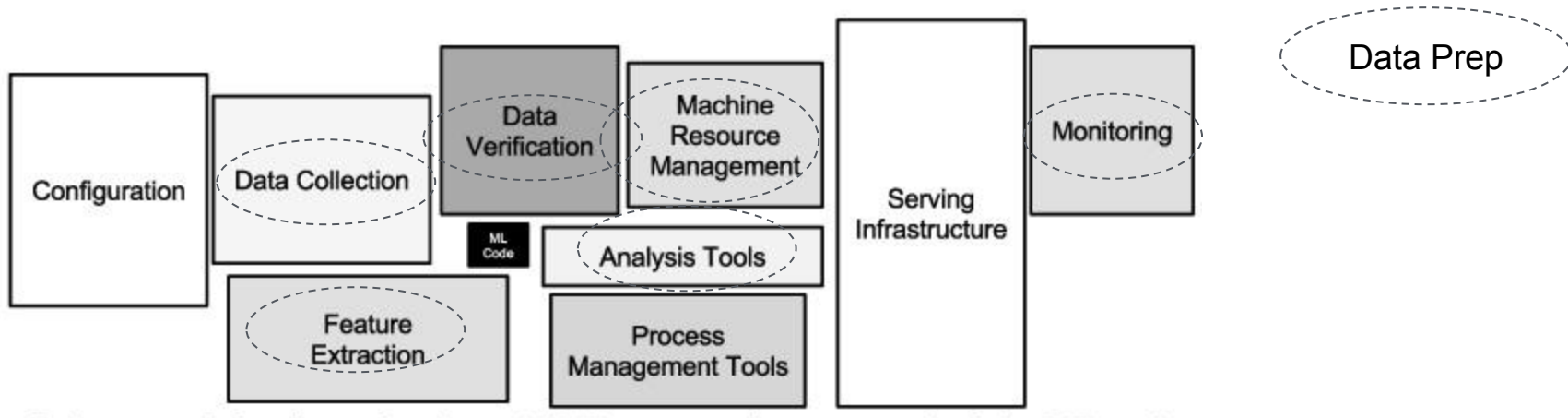
# Data Prep - Significance



*"Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small black box in the middle. The required surrounding infrastructure is vast and complex."*

**Paper from Google - NeurIPS 2015**

# Data Prep - Significance



*"Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small black box in the middle. The required surrounding infrastructure is vast and complex."*

**Paper from Google - NeurIPS 2015**

Implement, Operate, Audit, Access, Monitor Model Input and Output

# Data Prep for Models - Nature

---

- Also called Feature Engineering
- Features are variables generated from data
  - Continuous process (Batch + Near Realtime + Realtime)
- Large in number ('00s to '000s) & evolving
- Frequently executed

Customer	SKU	Name
17826162	0293192	Thai Dragon Fruit



Customer	Premium	Imported
17826162	15% of txns	5% of spend

Retail Customer  
(X GB)

Features  
(~X/1000)



# Data Prep Pipeline Consumers

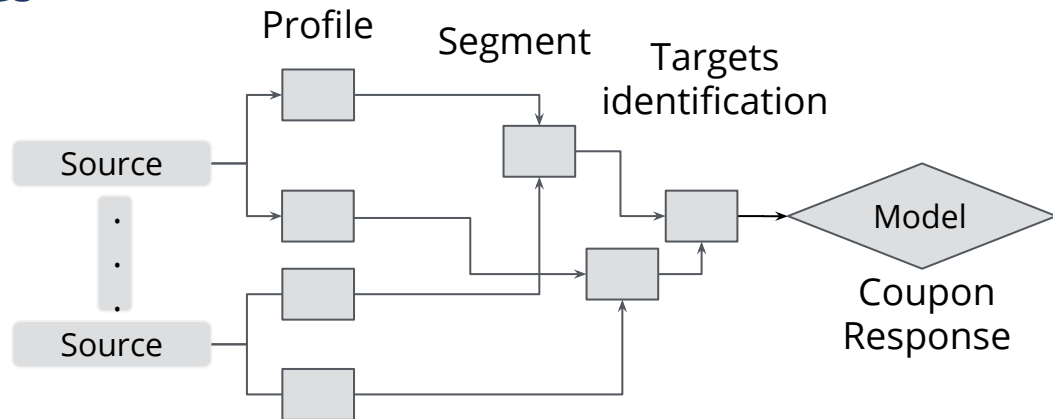
---

Nature	Example	Timing	Users
Models	Prediction	Batch or Realtime	Data savvy (python etc) Understand scale & contracts
Automation	Reordering	Typically batch	Application developers (Java) Less flexibility & More Contracts
Analysis	Segmentation	Adhoc	SQL-based tooling Explainability critical Availability and access focus Completeness nice to have

# Nature of the Problem

# Data Prep Pipelines - Structure

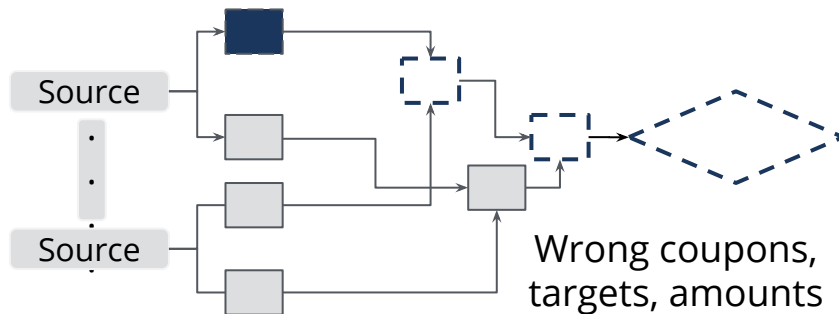
- Multiple, intersecting DAGs
- Can be broad and deep
- Continuous change
- Compute intensive
- Long execution times
- High volume of data



Sample: 50GB/day, 2M customers, 200 features, 10 pipelines, 4 hours execution time

# Network+Time Makes Everything Hard

- Fluid systems that evolve over time
- Change propagates thru network
- Validation is always incomplete
- Hidden dependencies
- Impact can't be undone



Sample: 6TB recompute and unknown \$\$ cost when profile is wrong!

# Where does time go?

# Where does time go? Stitching Systems

---

- Data sources, semantics, transformations span large space
- Rapid change in business need
- Natural fit for Python
- Flexibility and speed critical
  - Quick movement from test to prod
- Limited organizational resources
  - Robustness and productivity is critical

Python's advantage: Interfacing (sqlalchemy, REST etc), computation (pandas etc), application framework (django etc)

# Where does time go? xData

---

- Explanation for each feature & value
  - Three different languages/tests/contracts
  - Business/application consumers want to know
- Everyday, for every output
  - Many combinations - versions x runs x dependencies
- Reproducibility is a requirement
  - No explanation is credible without one

xData will enter conversations soon

# Where does time go? Changes

---

- Changes - Involuntary (bugs) and Voluntary (functionality)
  - Different classes of features with different behaviors
- Thousands of lines of dense code
  - Corner cases + large volumes of data
- Correctness issues can be **very** expensive
  - Embarrassment and \$\$\$
  - Laborious investigation, fixing code and data

Expect Python data management layer!



# Where does time go? Resource Management

---

- Pandas is memory intensive: 5x rule
  - Continuous optimization and careful coding
  - Explicit memory management
- Implementing tradeoffs
  - Dev speed (D), Ops cost (O), Scalability (S)
- Need more high perf data structures (lists, dicts)
- Hidden Gem - Itertoolz

# Required Capabilities

# Implement: Provide Structure to Development

- Modular class structure
  - Flexible configuration
  - Pre/post exec validation
  - Pytest integration
  - Automatic documentation
  - Data quality checking
- Productivity enhancers
  - Feature specification DSL
  - Query/other templates

```
===== 1 passed in 0.01 seconds =====
✓ Completed successfully
✓ Loaded imported the cars module
✓ Module has a provider attribute
✓ Able to instantiate the module
✓ Module has testdata
✓ Testdata appears valid
✓ Able to load test data
✓ Configured the module
✓ Validated the configuration
✓ Starting process
✓ Executed the process function
✓ Validated the results
✓ Stored the results
Results in /home/ubuntu/JohnWor

1 {
2   "schema": "user:default:v1",
3   "id": "user.none.completed_order",
4   "entity": "user",
5   "granularity": "NONE",
6   "name": "completed_orders",
7   "owner": "feast@example.com",
8   "description": "This feature rep",
9   "uri": "https://example.com/",
10  "valueType": "INT32",
11  "tags": [],
12  "options": {},
13  "dataStores": {
14    "serving": {
```

# Operate: Flexible & Controlled Execution Management

- Parameterization
  - Easily extensible
  - Dynamic defaults
- Notifications w/ callouts
- Automated deployment
  - Coordinated across modules
  - Impact analysis of changes
- Service integration
  - Prefect, Netdata, Supervisor

Enrich / Demo / SimpleFEPipeline / Run Arguments

SimpleFEPipeline

Pipeline to demonstrate feature engineering

This pipeline one or more arguments to run. Please specify them.

SimpleFE

dataset\*

1571294391\_AutoFE\_Sample

Name of Uploaded Dataset

idcol\*

CUSTOMER\_ID

Primary key of sample dataset

check\_p  
check\_r  
clean\_a  
cleanup  
cleanup  
cleanup  
clone\_r  
collect  
compile  
configure\_netdata  
create\_superuser  
create\_virtualenv  
datetime  
decrypt  
deploy\_server  
distribute\_sdk  
encrypt  
enrichcmd  
enrichcmd\_add\_enrich  
enrichcmd\_add\_keys  
enrichcmd\_add\_users  
enrichcmd\_check\_libs  
enrichcmd\_install\_envs  
enrichcmd\_install\_libs  
enrichcmd\_remove\_enrich  
enrichcmd\_setup  
enrichcmd\_show\_ssh  
flush\_cache  
full\_install  
full\_remove  
full\_upgrade

mlflow mlflow\_server STOPPED Not started

netdata netdata RUNNING pid 28616, uptime 5 days, 8:54:47

spark spark\_master STOPPED Not started

spark spark\_worker STOPPED Not started

[OPS] Show ssh commands

Flush OS cache.

Full install from scratch

State	Description
STOPPED	Not started
STOPPED	Not started
RUNNING	pid 17519, uptime 5 days, 7:47:00
RUNNING	pid 10197, uptime 5 days, 2:25:46

# Audit: Use and Manage Metadata Extensively

- Knowing what changes your data goes through
- End-to-end auditability
  - All data and all runs
  - Metadata standardization
- Discovery and reuse
  - Pipelines, modules
  - Lineage search
- Early warning systems
  - Input/output quality checks
  - Note critical decisions

<pre>    "release": "v0.3.5",   },   {     "date": "2018-03-08 09:38:56 +0530",     "commit": "7bdd63e7a86ed08768c07e1ef3ee3328a2899551",</pre>	
? Status	success
👤 By	pingali
🖨 Server	production.enrich.cndlabs.scribbledata.io
🕒 When	3 weeks, 3 days ago
⌚ Duration	1028s
⇄ Transforms Applied	MemberProfileSummary (v1.0:v1.0 by CnDLabs) FileOperations (v1.0:v1.0 by Builtin) SQLEXP (v1.0:v1.0 by Builtin) JSONSink (v1.0:v1.0 by Builtin) CampaignMeta (v1.0:v1.0 by Venkata Pingali) TableSink (v1.0:v1.0 by Builtin) MemberProfileFinalize (v1.0:v1.0 by CnDLabs)
🔗 Code	<a href="#">? enrich-assist - Data catalog and search interface</a> (Commit: 7bdd63e...) <a href="#">? enrich-nufuture - nuFuture Digital applications</a> (Commit: fee20c3...) <a href="#">? enrich-scribble - Core scribble applications</a> (Commit: 230e716...)

# Access: Stable, Safe, Continuous Consumption

- Marketplace for data discovery
- Data contracts
- Isolation: Multi-tenant namespaces
  - File system, tables, S3
- Time: Versioned namespaces
  - Storage locations
  - Metadata
- Linked data and code

anonymized\_member\_profile (876617 x 457)

**Description** Consolidated and anonymized member profile

**Notes**  
[Final] Members: 876617  
[Final] Unique Days: 165  
[Final] First Day: 20180604  
[Final] Last Day: 20181123  
[Files] Incomplete: 0  
[Files] Errors: 0  
[Files] Empty: 0  
[Files] Included: 184  
[Cleaning] Dropped amtsold < 0 for simplicity  
[Cleaning] Some transactions dropped due to incomplete product master  
[Cleaning] Some transactions dropped due lack of mapping info in txn\_customer

**Path** nufuture/DataModels/output/FE-Loyalty-Finalize/features-finalize-20181126-164144/anonymized\_member\_profile.csv

**Shape**

**Component**

**Feature Marketplace** API Back  
Status of Features and Models

**Overview** **Datasets** **Features** **Models** **Requests**

**Overview**  
Summary of the features and models, and requests for new features

3 Datasets	235 Features	0 Models
0 Requests		

# Takeaways

---

- Data prep required for all ML
  - Costly, cumbersome, error prone
  - Structure of the problem makes it hard
- Provide support in all stages of lifecycle
  - Implement, operate, audit, and consume
- xData will grow
  - Model correctness q's are often data correctness q's

THANK YOU  
FOR YOUR TIME

---

DENVER  
Littleton



BANGALORE  
Indiranagar | HSR



[pingali@scribbledata.io](mailto:pingali@scribbledata.io)