cloudera
Ask Bigger Questions

End-to-end analytics with Apache Spark

Sandy Ryza

Me

- Data scientist at Cloudera
- Recently lead Apache Spark development at Cloudera
- Before that, committing on Apache Hadoop
- Before that, studying combinatorial optimization and distributed systems at Brown

Large Scale Learning

What for?

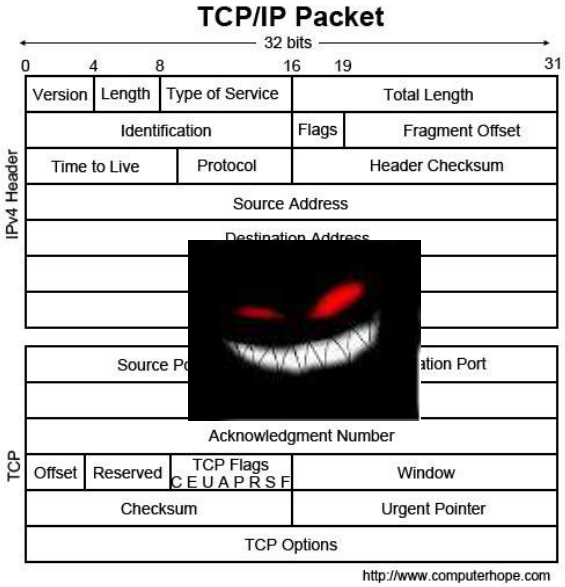
Detect Things That Will Go Wrong

- Churn prediction
- Detect machine failures

Identify Bad Actors

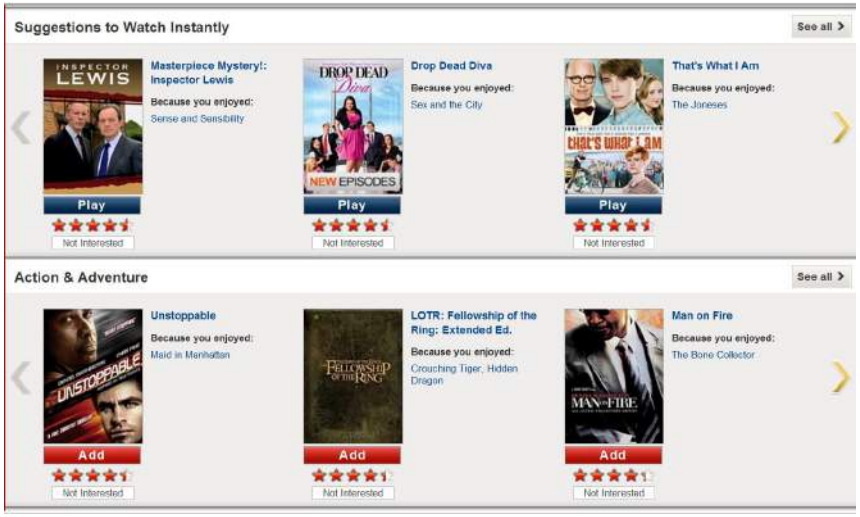
Identify Bad Actors

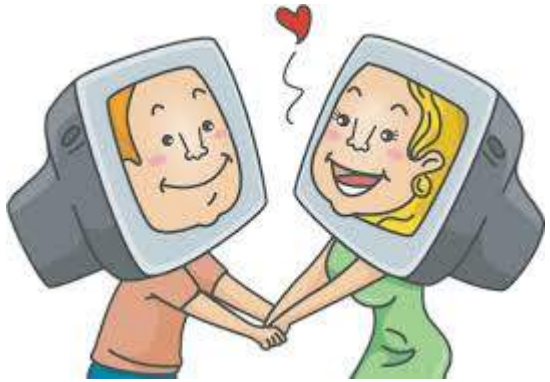
- Network intruders
- Payment fraudsters
- Adversarial advertisers
- Insurance claim grifters



Provide Recommendations

- Movies to stream
- Music to stream
- Products to buy
- Ads to serve
- People to date





The Lab and the Factory

The Lab

- Question-driven
- Interactive
- Fixed data
- Output -> report or in-database scoring engine

The Factory

- Metric-driven
- Automated
- Fluid data
- Output -> production system that makes customer facing decisions

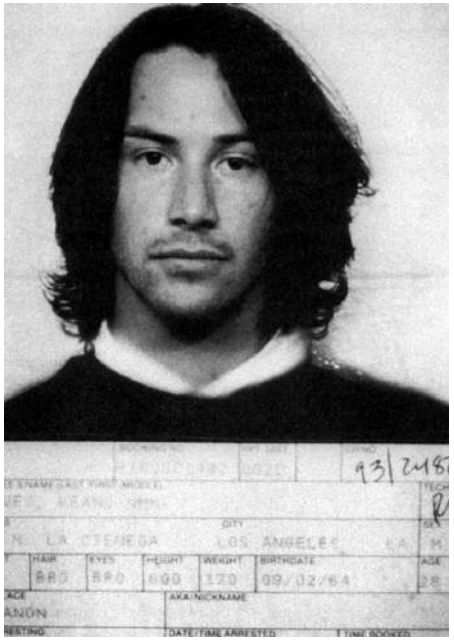
What does it mean to productionize your machine learning?

Some models can be safely applied in batch

- Run your churn predictor every day and act on it at night

Most use cases need real time serving

- Catch bad actors before they do bad stuff



- Make recommendations upon site visit

Recommendations need real time updates

Infrastructure

Model Building

Model Serving

Model Updating

Oryx

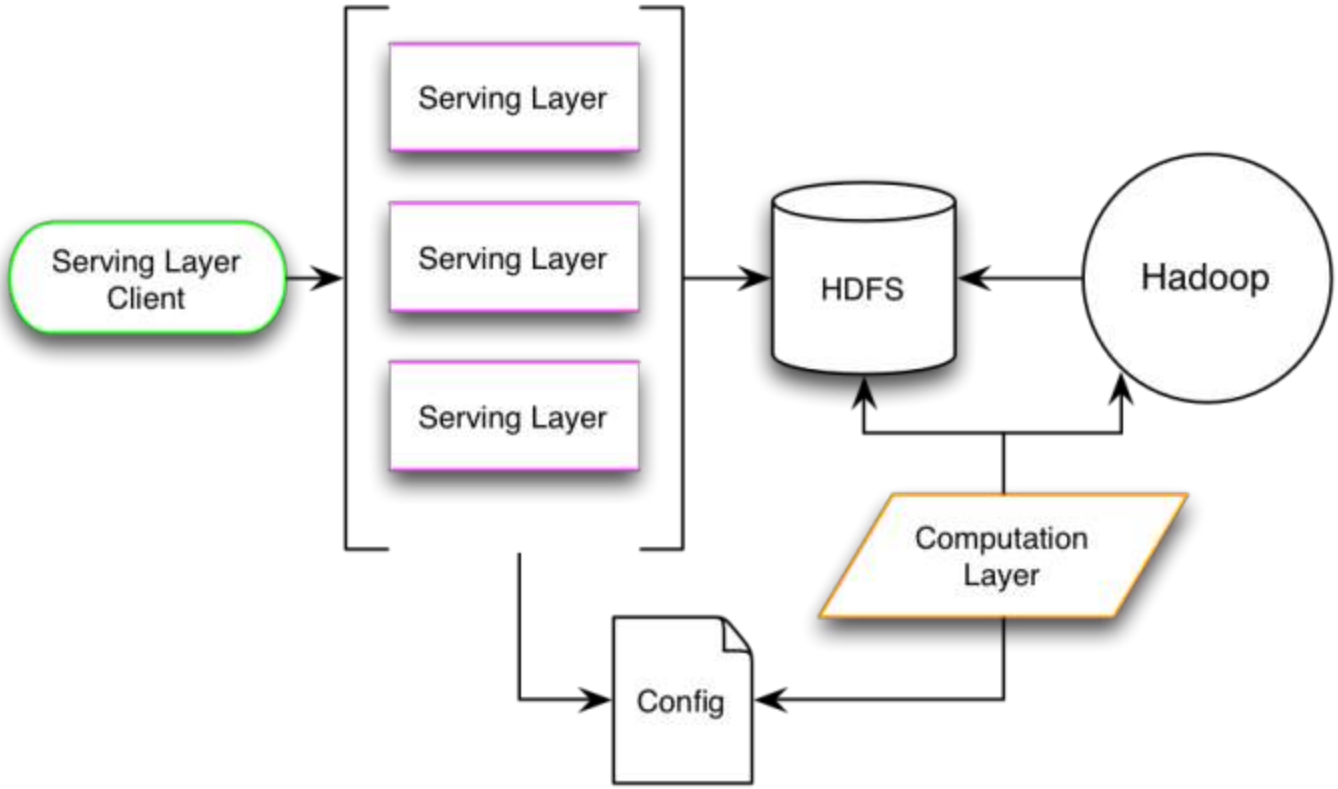
Oryx

- https://github.com/cloudera/oryx
- Focused on building real-time applications using machine learning
- Model building and model serving infrastructure
- Model serving consumes PMML
- Most common use is recommendation

Oryx 1.0

- Model building
  - Custom MapReduce algorithms
- Model update
  - Partitioned by user
  - Local to each serving daemon

Oryx 1.0


Serving Layer
Serving Layer
Serving Layer
Serving Layer Client
HDFS
Hadoop
Computation Layer
Config

Algorithms - one of each

- Recommendation
  - Alternating least squares for collaborative filtering
- Classification
  - Random decision forests
- Clustering
  - K-means

MLLib

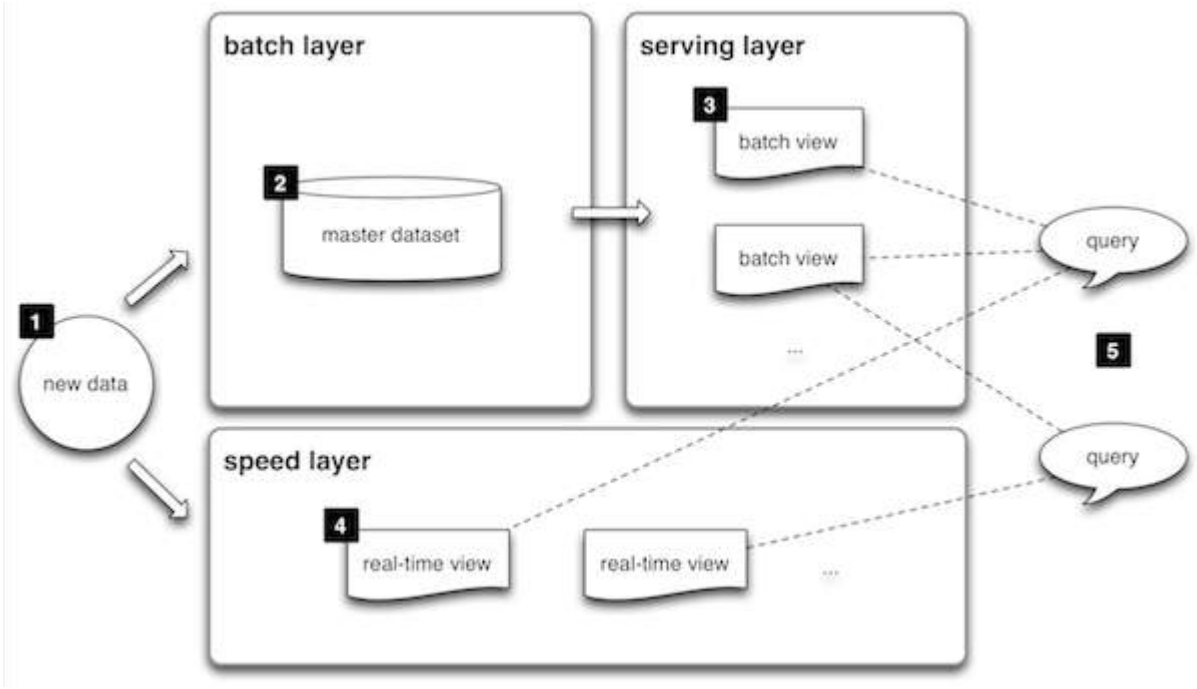
	Discrete	Continuous
Supervised	**Classification** ● Logistic regression (and regularized variants) ● Linear SVM ● Naive Bayes ● Random Decision Forests (soon)	**Regression** ● Linear regression (and regularized variants)
Unsupervised	**Clustering** ● K-means	**Dimensionality reduction, matrix factorization** ● Principal component analysis / singular value decomposition ● Alternating least squares

Oryx 2.0

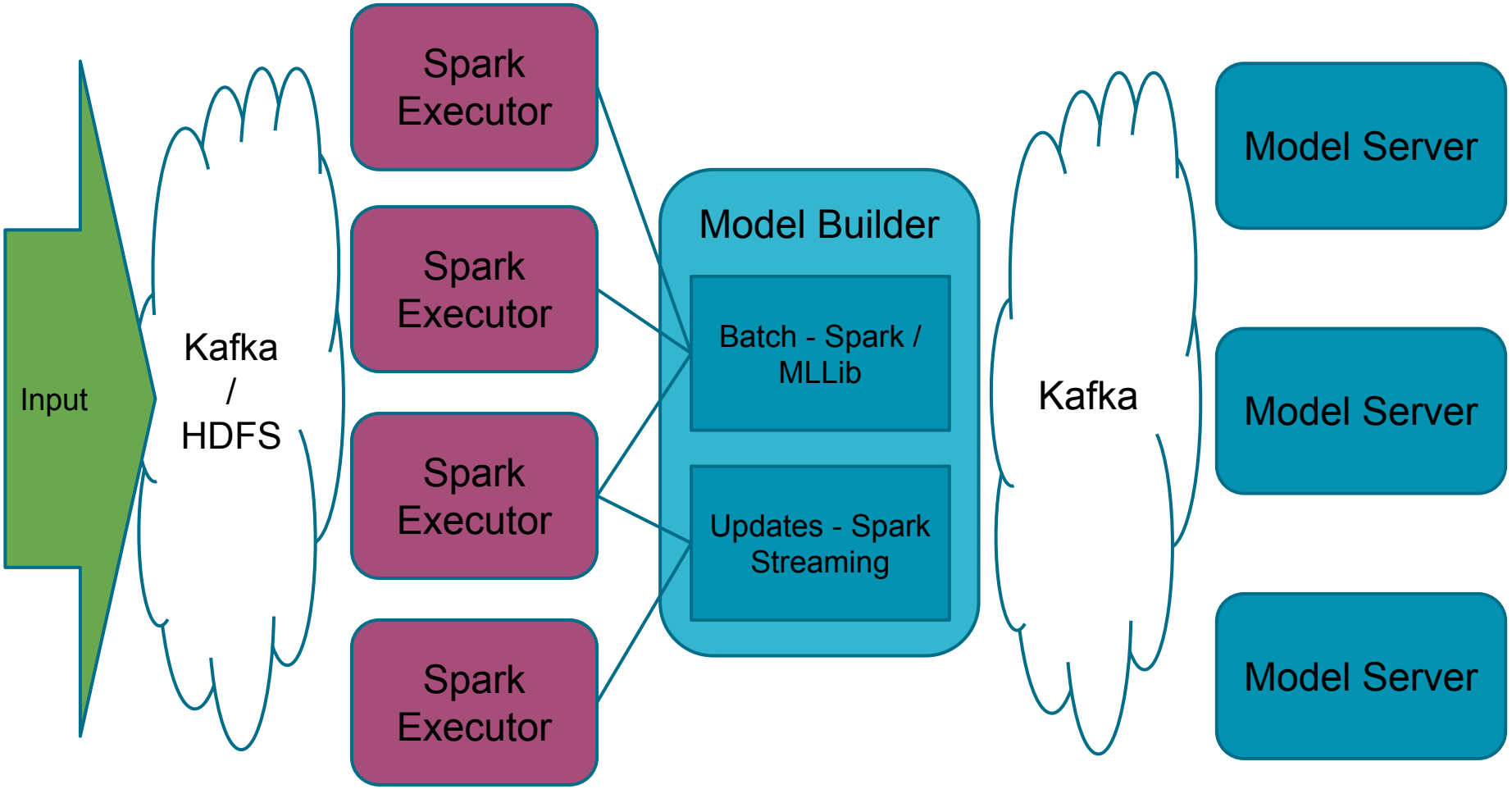
- Replace MR algorithms with MLLib
- Replace real-time update with Spark streaming

“Lambda Architecture”

- Periodically train on whole data
- Incremental updates with new data



Oryx 2.0


Input
Kafka
/
HDFS
Spark
Executor
Spark
Executor
Spark
Executor
Spark
Executor
Model Builder
Batch - Spark /
MLLib
Updates - Spark
Streaming
Kafka
Model Server
Model Server
Model Server

What could go into MLLib?

- PMML output
- Model update
- Hyper-parameter tuning

Contributions?

- https://github.com/cloudera/oryx