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
Algorithms - one of each

- Recommendation
  - Alternating least squares for collaborative filtering
- Classification
  - Random decision forests
- Clustering
  - K-means
<table>
<thead>
<tr>
<th></th>
<th>Discrete</th>
<th>Continuous</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Supervised</strong></td>
<td><strong>Classification</strong></td>
<td><strong>Regression</strong></td>
</tr>
<tr>
<td></td>
<td>● Logistic regression (and regularized variants)</td>
<td>● Linear regression (and regularized variants)</td>
</tr>
<tr>
<td></td>
<td>● Linear SVM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>● Naive Bayes</td>
<td></td>
</tr>
<tr>
<td></td>
<td>● Random Decision Forests (soon)</td>
<td></td>
</tr>
<tr>
<td><strong>Unsupervised</strong></td>
<td><strong>Clustering</strong></td>
<td><strong>Dimensionality reduction, matrix factorization</strong></td>
</tr>
<tr>
<td></td>
<td>● K-means</td>
<td>● Principal component analysis / singular value decomposition</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● Alternating least squares</td>
</tr>
</tbody>
</table>
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
What could go into MLLib?

- PMML output
- Model update
- Hyper-parameter tuning
Contributions?

- [https://github.com/cloudera/oryx](https://github.com/cloudera/oryx)