Machine Learning Model Life Cycle Management
In Production - #aiops #mlops
Who are we?

Harish Doddi  
CEO

- Lyft Surge Pricing Model
- Twitter’s Distributed Photo Storage - 12 PB
- Architected Snapchat Stories - scaled from 0 to billions

Jerry Xu  
CTO

- Lyft ETA Machine Learning Model - replaced Google
- Twitter’s Manhattan - replaced Cassandra
- Founding team of Windows Azure

Team: Previously worked at places like Amazon, Microsoft, and Anaconda

Headquarters: 350 Townsend Street, Suite 204, San Francisco, CA
Today’s Enterprises Data Science Lifecycle

Development
- Data Science
- Initial Analysis
- Model development using multiple frameworks
- Training and Experimenting

Production
- Engineering
- Deployment
- Workflow process
- Roll back strategy
- A/B Testing
- DevOps
- Model Performance Monitoring
- Infrastructure Monitoring
- Scaling
- Model Governance

Post-Production
- DevOps
Production and Post-Production Deployment, Monitoring & Governance

Data Aggregation

Discovery & Analysis

Training & Experimenting
Lesson 1

Need for single centralized production team and platform
Financial Institutions Production Environment

- Fraud Data Science
  - Production infrastructure
  - Databricks

- Marketing Data Science
  - Production infrastructure
  - Python
  - Jupyter
  - DataRobot

- Credit Risk Data Science
  - Production infrastructure
  - Java
  - Dataiku
  - H2O.ai

Production infrastructure
Hidden Technical Debt in Machine Learning Systems

Google Paper

Hidden Technical Debt in Machine Learning Systems

Machine learning offers a fantastically powerful toolkit for building complex systems quickly. … it is remarkably easy to incur massive ongoing maintenance costs at the system level when applying machine learning.

- Boundary erosion
- Entanglement
- Hidden feedback loops
- Undeclared consumers
- Data dependencies
- Changes in the external world
- System-level anti-patterns
Production Model Deployment

- Fraud Data Science
- Marketing Data Science
- Credit Risk Data Science

Horizontal DEVOPS AI team
Centralized Production AI Platform like Datatron
Lesson 2

Start adopting best practices and standardization early
Old Way

Teams face cross-functional inefficiencies

- Teams operate in silos, don’t speak the same language
- Errors due to lack of communication
- Engineering has to write stand-alone scripts
New Way
Central devops platform for AI models streamlines the process and reduces the inefficiencies.
Lesson 3

Data Science Universe shouldn’t be limited
Old Way
No Model Containerization
Data Scientist Universe is limited

Team → Team → Team → Model

× × ×

New Way
Model Containerization
Data Scientist Universe is un-limited

Team → Team → Team → New Frameworks

Model
TensorFlow
scikit-learn
Apache Spark

Upcoming Frameworks
Lesson 4

Models may go wrong, you need to monitor them continuously
South Park and Alexa
Business is **continuously losing** value

Minimize the continuous loss
Production Model Monitoring
Allows You To Act Preemptively

Old Way

Post-mortem reports

New Way

# of Fraud Transactions

Jan  Feb  Mar  Apr  May  Jun

Reality
Model Prediction
Anomalies Detected
Monitoring for Machine Learning Models

- **Model Performance monitoring**
  - Confusion Matrix
  - Gain and Lift charts
  - Kolomogorov Smirnov chart
  - Area Under the ROC curve
  - Gini Coefficient
  - Concordant – Discordant ratio
  - Root Mean Squared Error (RMSE)
  - etc

- **Model Timeout monitoring**
- **Infrastructure monitoring**
- **Organization KPI monitoring**
- **Anomaly monitoring**
Lesson 5

You either NEVER deploy a model, or you have to do it over and over again.
ML model is a continuously optimizing process

- Concept drift
- New concept comes up
Connecting Machine Learning to Software world

<table>
<thead>
<tr>
<th>Before</th>
<th>Now</th>
<th>Future</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software deployment</td>
<td>Software deployment every day</td>
<td>Machine Learning models will deploy very frequent and fast</td>
</tr>
<tr>
<td>once a 1 or 2 years</td>
<td>BUT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Machine Learning models deploy very slow</td>
<td></td>
</tr>
</tbody>
</table>
Lesson 6

Model Governance should be automated as much as possible
Production Model Governance
Each Model Action Is Logged

Old Way

⚠️ Time for an audit ⚠️

Log
Comments
Version
Log
Log
Log
Log

New Way

Model Versioning

Log
Comments
Versions

🌟🌟🌟
Lesson 7

Be prepared, your number of model will grow
Deployment Learning: 1 model vs Multiple models

Growing Use of Deep Learning at Google

Number of directories containing model description files

Unique Project Directories

Across many products/areas
- Apps
- Maps
- Photos
- Gmail
- Speech
- Android
- YouTube
- Translation
- Robotics Research
- Image Understanding
- Natural Language Understanding
- Drug Discovery
Value Proposition: Cost per order of magnitude

Without Model Management

As the number of models increases, the cost also increases

With Model Management

As the number of models increases, the cost significantly decreases
Lesson 8

Senior people are required at later stages
Software Development vs ML Model Development

Senior People

Requirements → Design → Implementation → Testing → Evolution

Senior People

Requirements → Data Preparation → Training / Testing → Deploy to Production → Monitoring and Optimization
Lesson 9

Your real work starts AFTER you deploy the model to production
Enterprise AI Life Cycle

Exploration → Training → Deploy

X
Enterprise AI Life Cycle After Deployment

- **Deploy**
  - Blue-Green deployment
  - Rollback
  - Canary

- **A/B Testing**
  - Split traffic
  - Shadowing
  - Multi-Armed Bandit
  - Optimization

- **SLA**
  - Model timeout
  - Fall back strategy
  - Alerting

- **Monitoring**
  - Model performance
  - Model latency
  - Infrastructure monitoring

- **ML Frameworks Support**
  - Tensorflow, sklearn, H2O, SAS, SparkML, etc.

- **IT Env Integration**
  - Cluster management integration
  - Security check

- **Auditing**
  - Versioning
  - History
  - Approval process

- **Model Selection**
  - Model routing
  - Challenger
  - KPI based selection

- **Anomaly Detection**
  - Feature distribution
  - Model result

- **Troubleshooting**
  - Replay
  - End-to-end tracing
  - Logging