Nexus Feature Store
powering Disney Magic

Dustin Hamerla, Manager ML Engineering, Disney Streaming
Agenda

• Nexus Feature Store
  • Overview & Motivation
  • Capabilities
  • Architecture
Overview & Motivation
ML Platform overview

Democratize & accelerate scaleable ML development across Disney Streaming
Why do we need a Feature Store?

*Improve feature creation (Data Engineers) and feature consumption (Data Scientists) for various ML use cases*

- **Break-up bespoke** pipelines
- **Decouple** feature engineering from model development
- **Support** various computing environments
- **Discover & share** features
Why do we need to build our own Feature Store?

- Global scale batch training & scoring
  - Region-specific data
  - Focus on experimentation
  - Feeding multiple datasets – not just “one training dataset”
  - Bridge between Spark based dataprep and single-node training & scoring

- Customized data model
  - Compatibility with larger ML platform

- Existing ML-platform infrastructure and teams
Nexus Feature Store Capabilities
Nexus Feature Store capability map

<table>
<thead>
<tr>
<th>COMPUTE/ WRITE FEATURES</th>
<th>Feature Store UI</th>
<th>Feature Store internal compute</th>
<th>Auth. &amp; User rights</th>
</tr>
</thead>
<tbody>
<tr>
<td>Batch (write)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nearline (write)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Real-time (compute)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**READ FEATURES**

- **Batch**
  - Read Spark
  - Read Non-Spark
  - Feature transformation UDF applied in Spark
  - Batch processing, sync to online
  - Stream processing, ingestion to offline store
  - Feature transformation UDF applied in real-time

- **Real-time**
  - Internal data distribution engine
  - Stream processing, ingestion to online store

**Feature Statistics**
**Feature Validations**
**Feature Store UI**
**Feature Store mechanism**
**Feature Store mechanism**
**Sandbox environment**
**Feature transformations**
**Client SDKs**
Nexus Feature Store Architecture
Nexus FS Architecture - Spark write & Non-Spark read

**Batch Processing (Spark jobs)**
- Batch Data (S3, Snowflake, ...)
- Offline Store (Delta Lake-S3, Meta store)

**Feature Store**
- Feature Registry
- Online Feature Service
- Online Store (Redis cluster)
- “FeatureGroup” level
  - Users: Data Engineers
- “FeatureService” level
  - Users: Data Scientists

**Write**
- get FeatureStore
- get FS.FeatureGroup(s)
- create FS.FeatureService
- ingest data from one or many FeatureGroups to a FeatureService

**Read**
- get FS.FeatureStore
- get FS.FeatureService
- load subsets of data from FeatureService (e.g. sub-sampled, sub-partitioned etc.)

**Metadata DB**

**SDK**

**Online Feature Service**

**Actual Feature Data**

**Metadata**

**Users: Data Engineers**

**Users: Data Scientists**
Nexus FS Architecture - Streaming write & real-time read

**Write**
- get FeatureStore
- get FS.FeatureGroup
- ingest streaming data to online (& offline) FeatureGroup

**Read**
- get FeatureStore
- get FeatureGroup(s)
- Define FeatureGroup joins in a QueryPlan
- Create FeatureService and save QueryPlan
- Get FeatureVector in milliseconds from FeatureService by passing primary-keys from real-time request
Thank you!