Feature Store: the missing data layer in ML pipelines?¹

Kim Hammar
kim@logicalclocks.com
@KimHammar1
February 1, 2019

Data is the hardest part of ML and the most important piece to get right.

- Uber

Jeremy Hermann and Mike Del Balso. *Scaling Machine Learning at Uber with Michelangelo.*
“Data is the hardest part of ML and the most important piece to get right.

Modelers spend most of their time selecting and transforming features at training time and then building the pipelines to deliver those features to production models.”

- Uber²

---

“Data is the hardest part of ML and the most important piece to get right.

Modelers spend most of their time selecting and transforming features at training time and then building the pipelines to deliver those features to production models.”

- Uber³

A feature store is a central vault for storing documented, curated, and access-controlled features.

The feature store is the interface between data engineering and data model development.
The feature store enables:
- Reusability of features between models and teams
- Automatic backfilling of features
- Automatic feature documentation and analysis
- Feature versioning
- Standardized access of features between training and serving
- Feature discovery
Reusing Features
Without a Feature Store is Complex

Data Sources
Dataset 1 \arrow{downarrow} \quad Dataset 2 \arrow{downarrow} \quad \ldots \quad Dataset n \arrow{downarrow}

Siloed Feature Sets
Without a feature store it is typical to have feature sets stored in isolation from each other.

Models
Models are trained using sets of features. Without a feature store each model typically defines its own feature definitions, without feature sharing across models.
Reusing Features
With a Feature Store is Simple

Data Sources
- Dataset 1
- Dataset 2
- \ldots
- Dataset \( n \)

Feature Store
A data management platform for machine learning.
The interface between data engineering and data science.

Models
Models are trained using sets of features.
The features are fetched from the feature store and can overlap between models.
The Components of a Feature Store

- **The Storage Layer:** For storing feature data in the feature store
- **The Metadata Layer:** For storing feature metadata (versioning, feature analysis, documentation, jobs)
- **The Feature Engineering Jobs:** For computing features
- **The Feature Registry:** A user interface to share and discover features
- **The Feature Store API:** For writing/reading to/from the feature store

<table>
<thead>
<tr>
<th>Feature Registry</th>
<th>API</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature Metadata</td>
<td>Jobs</td>
</tr>
<tr>
<td>Feature Storage</td>
<td></td>
</tr>
</tbody>
</table>
Reading from the Feature Store:

```python
from hops import featurestore
features_df = featurestore.get_features(['average_attendance', 'average_player_age'])
```

Writing to the Feature Store:

```python
from hops import featurestore
raw_data = spark.read.parquet(filename)
pol_features = raw_data.map(lambda x: x**2)
featurestore.insert_into_featuregroup(pol_features, 'pol_featuregroup')
```
Summary

- Machine learning comes with a high technical cost
- Machine learning pipelines need proper data management
- A **feature store** is a place to store curated and documented features
- The feature store serves as an interface between feature engineering and model development, it can help disentangle complex ML pipelines
- *Hopworks*\(^4\) provides the world’s first open-source feature store

---


\(^5\) Thanks to Logical Clocks Team: Jim Dowling, Seif Haridi, Theo Kakantousis, Fabio Buso, Gautier Berthou, Ermias Gebremeskel, Mahmoud Ismail, Salman Niazi, Antonios Kouzoupis, Robin Andersson, and Alex Ormenisan.
References

- Hopsworks’ feature store\textsuperscript{6} (the only open-source one!)
- Uber’s feature store\textsuperscript{7}
- Airbnb’s feature store\textsuperscript{8}
- Comcast’s feature store\textsuperscript{9}
- GO-JEK’s feature store\textsuperscript{10}
- HopsML\textsuperscript{11}
- Hopsworks\textsuperscript{12}


