

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

Hopworks Hands On - Palo Alto

Kim Hammar

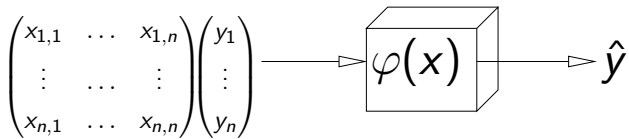
kim@logicalclocks.com

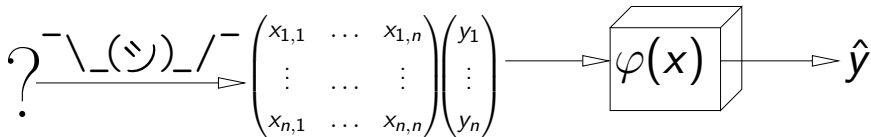
April 23, 2019



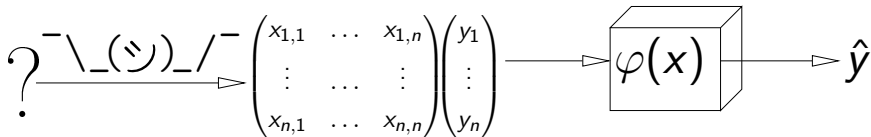
LOGICAL CLOCKS

¹Kim Hammar and Jim Dowling. *Feature Store: the missing data layer in ML pipelines?*
<https://www.logicalclocks.com/feature-store/>. 2018.





²Jeremy Hermann and Mike Del Balso. *Scaling Machine Learning at Uber with Michelangelo*.
<https://eng.uber.com/scaling-michelangelo/>. 2018.

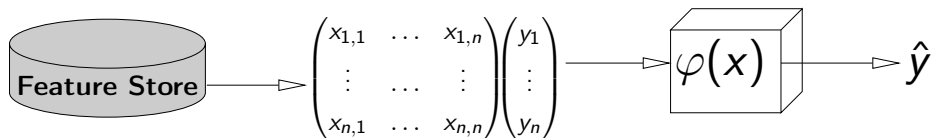


"Data is the hardest part of ML and the most important piece to get right."

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

²Jeremy Hermann and Mike Del Balso. *Scaling Machine Learning at Uber with Michelangelo*. <https://eng.uber.com/scaling-michelangelo/>. 2018.



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

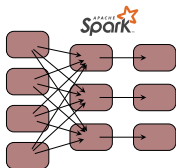
Modelers spend most of their time selecting and transforming features at training time and then building the pipelines to deliver those features to production models."

- Uber³

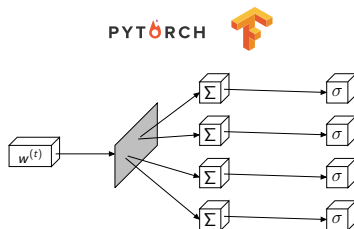
³Jeremy Hermann and Mike Del Balso. *Scaling Machine Learning at Uber with Michelangelo*. <https://eng.uber.com/scaling-michelangelo/>. 2018.

Merging Our Data Intensive and Compute Intensive Workloads

Data Intensive

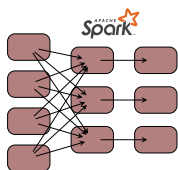


Compute Intensive

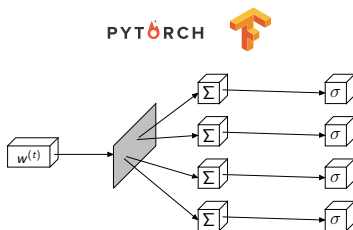


Merging Our Data Intensive and Compute Intensive Workloads

Data Intensive



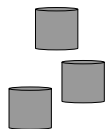
Compute Intensive



- 1 **What** is a Feature Store
- 2 **Why** You Need a Feature Store
- 3 **How** to Build a Feature Store (Hopsworks Feature Store)
- 4 Demo

Solution: Disentangle ML Pipelines with a Feature Store

Raw/Structured Data



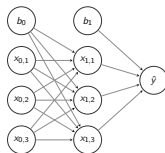
Feature Engineering



Training

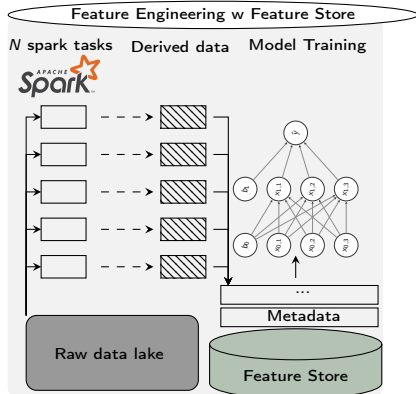
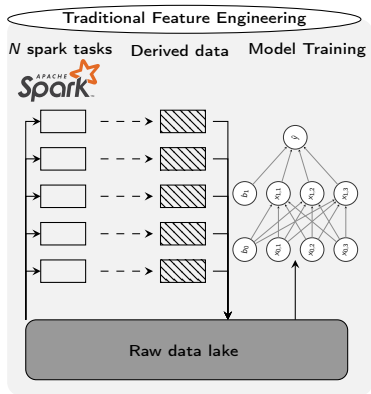


Models



- 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

Make ML-Features A First-Class Citizen in Your Data Lakes



- Make your features first-class citizens:
 - Document features
 - Version features
 - Invest in a data layer specifically for features (feature store)
 - Make features access-controlled and searchable

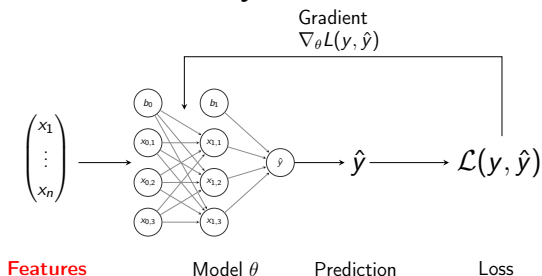
What is a Feature?

A feature is a measurable property of some data-sample

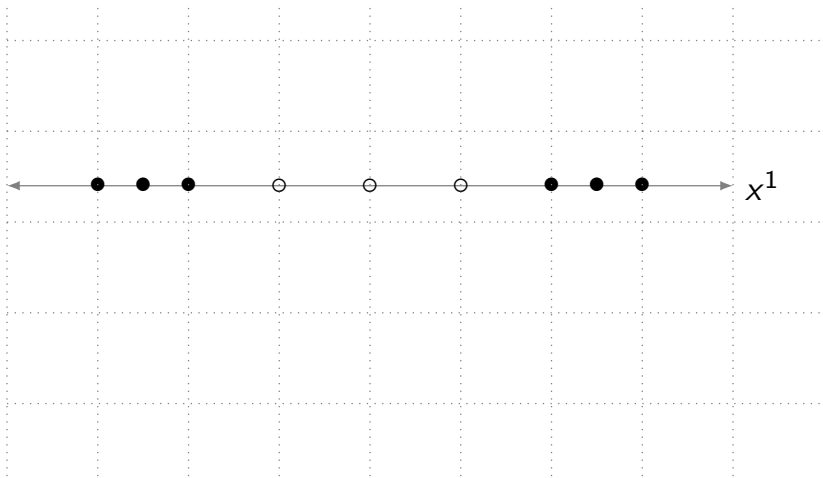
A feature could be..

- An aggregate value (min, max, mean, sum)
- A raw value (a pixel, a word from a piece of text)
- A value from a database table (the age of a customer)
- A derived representation: e.g an embedding or a cluster

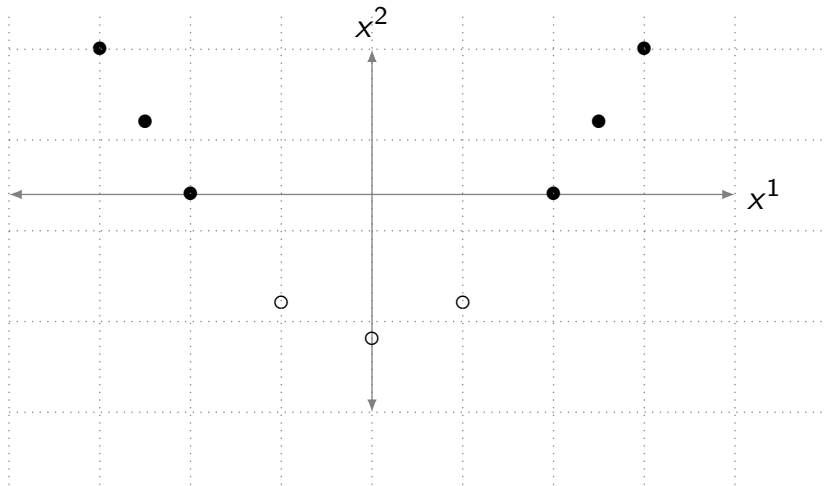
Features are the fuel for AI systems:



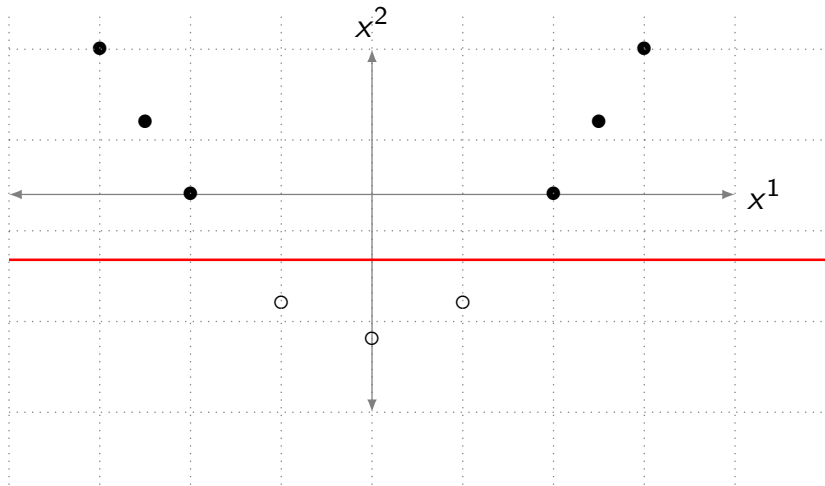
Feature Engineering is Crucial for Model Performance



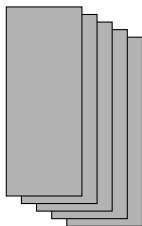
Feature Engineering is Crucial for Model Performance



Feature Engineering is Crucial for Model Performance

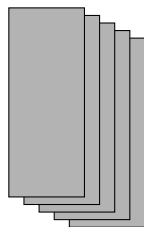


Feature Engineering is Complex

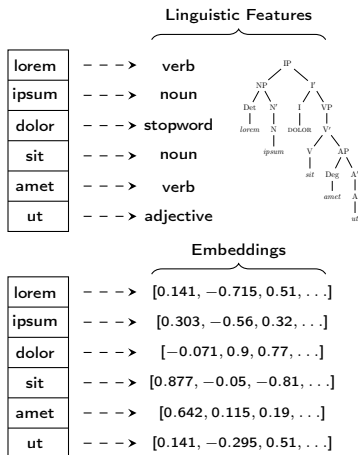


Input Data

Feature Engineering is Complex

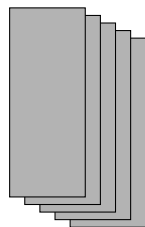


Input Data

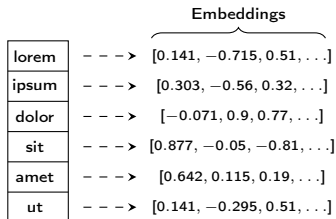
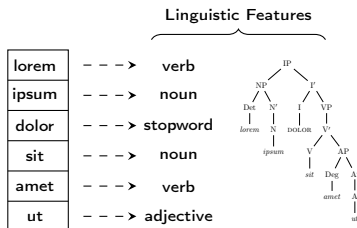


Feature Engineering

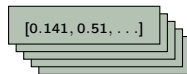
Feature Engineering is Complex



Input Data

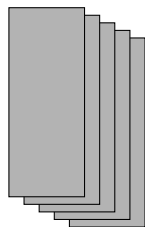


Feature Engineering

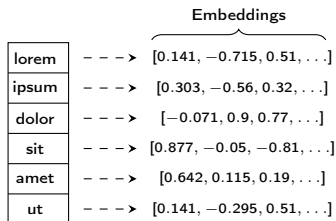
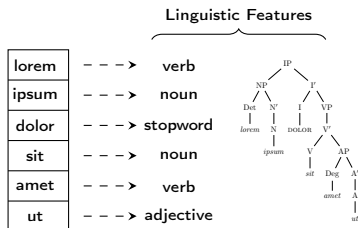


Feature Matrix

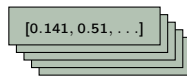
Feature Engineering is Complex



Input Data



Feature Engineering

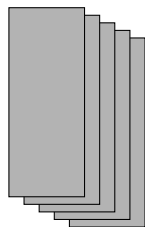


Feature Matrix

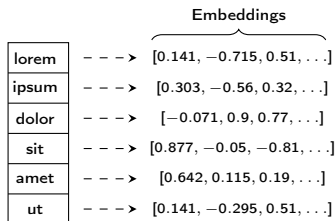
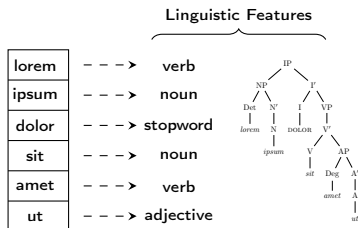


Train/Val/Test Split

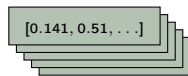
Feature Engineering is Complex



Input Data



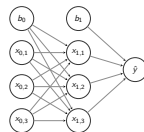
Feature Engineering



Feature Matrix

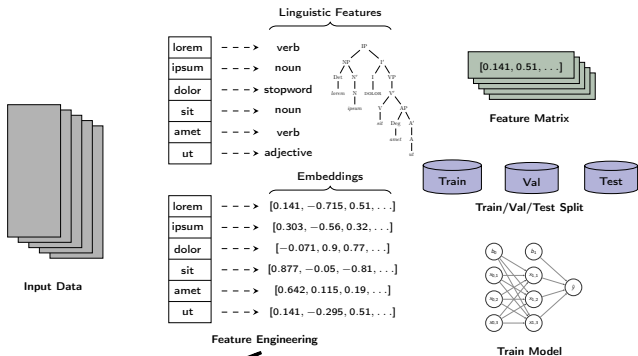


Train/Val/Test Split



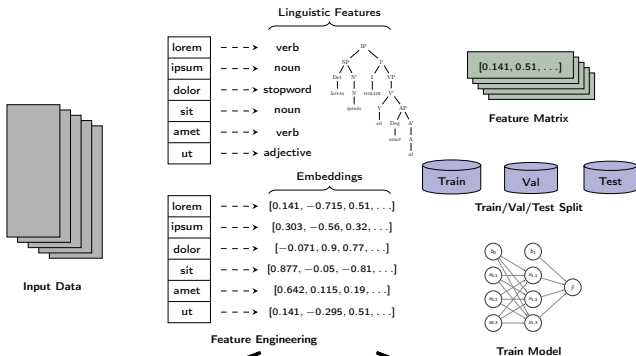
Train Model

Feature Engineering is Complex



How do you make this scale?

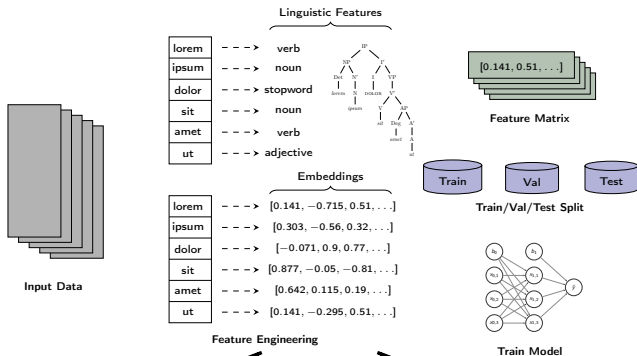
Feature Engineering is Complex



How do you make this scale?

How to manage the feature pipelines?

Feature Engineering is Complex

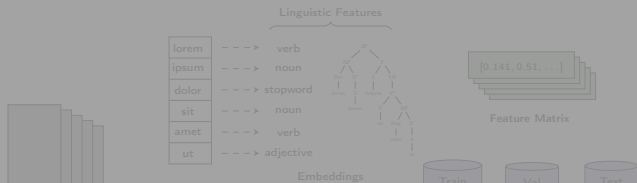


How do you make this scale?

How to manage the feature pipelines?

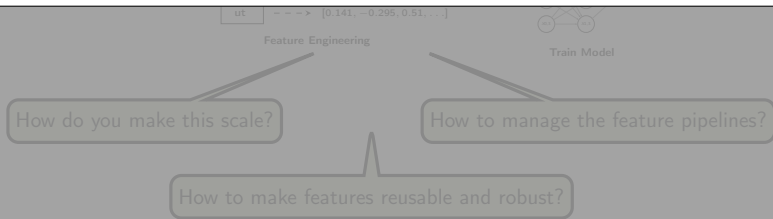
How to make features reusable and robust?

Feature Engineering is Complex



Feature Engineering is Complex Yet Crucial for Model Performance

Treat your features accordingly!

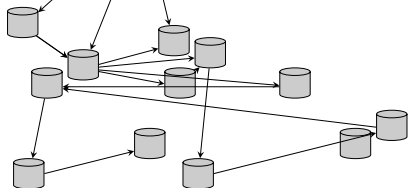


Feature Pipeline Jungles

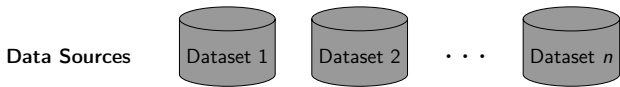
Data Lake (Raw/Structured Data)



Feature Data (Derived Data)

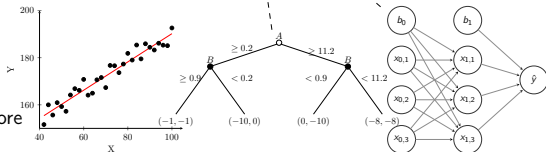


Disentangle Your ML Pipelines with a Feature Store

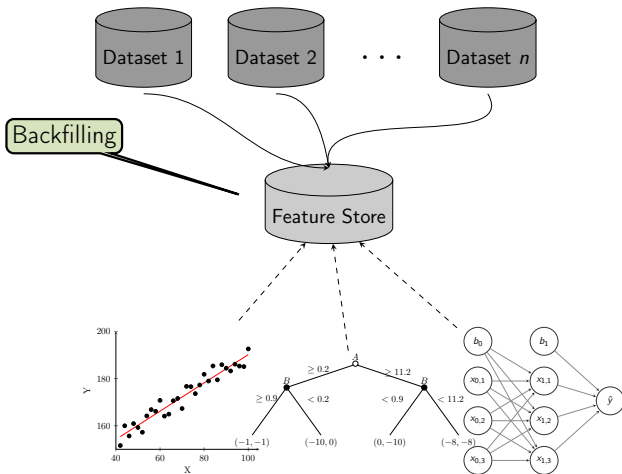


Feature Store
A data management platform for machine learning.
The interface between data engineering and data science.

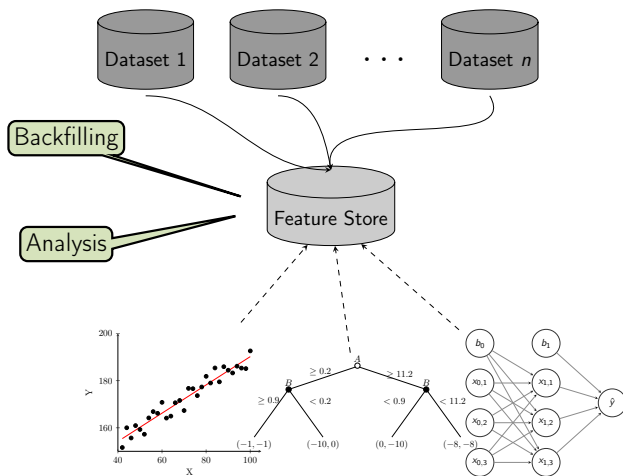
Models
Models are trained using sets of features.
The features are fetched from the feature store
and can overlap between models.



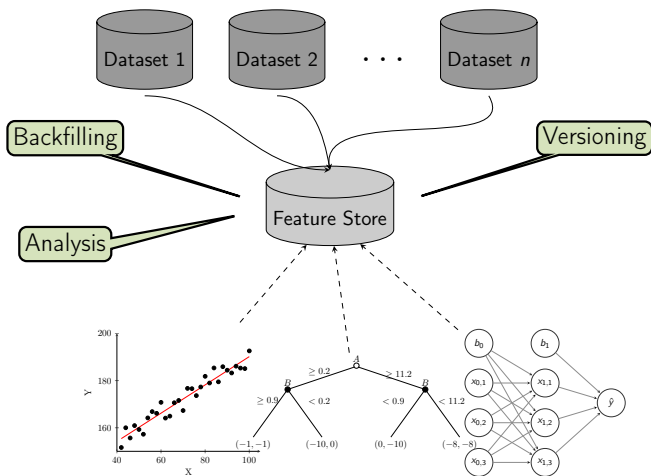
Disentangle Your ML Pipelines with a Feature Store



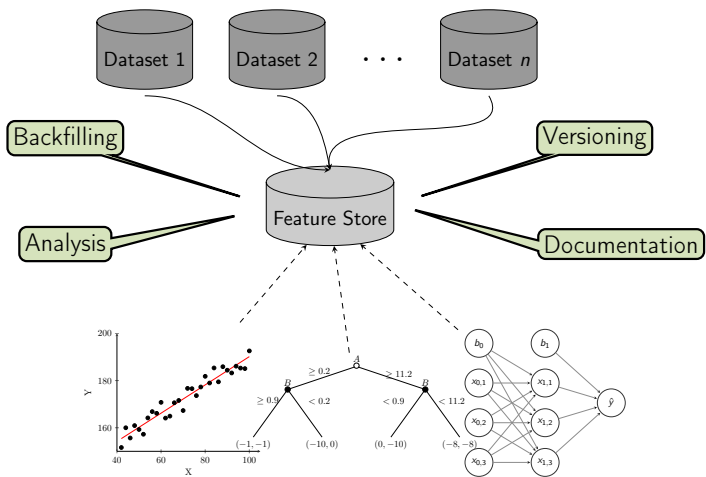
Disentangle Your ML Pipelines with a Feature Store



Disentangle Your ML Pipelines with a Feature Store



Disentangle Your ML Pipelines with a Feature Store



High-Level APIs and Abstractions

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

```
featurestore.create_featuregroup(
    f_df, "t_features",
    description="...", version=2)
```

```
d_dir = featurestore.get_training_dataset_path(td_name)
tf_schema = featurestore.get_tf_record_schema(td_name)
```

High-Level APIs and Abstractions

Read from the feature store

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

featurestore.create_featuregroup(
    f_df, "t_features",
    description="...", version=2)

d_dir = featurestore.get_training_dataset_path(td_name)
tf_schema = featurestore.get_tf_record_schema(td_name)
```

High-Level APIs and Abstractions

Read from the feature store

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

Write to the feature store

```
featurestore.create_featuregroup(
    f_df, "t_features",
    description="...", version=2)

d_dir = featurestore.get_training_dataset_path(td_name)
tf_schema = featurestore.get_tf_record_schema(td_name)
```


High-Level APIs and Abstractions

Read from the feature store

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

Write to the feature store

```
featurestore.create_featuregroup(
    f_df, "t_features",
    description="...", version=2)
```

Metadata operations

```
d_dir = featurestore.get_training_dataset_path(td_name)
tf_schema = featurestore.get_tf_record_schema(td_name)
```

Existing Feature Stores

- Uber's feature store⁴
 - Airbnb's feature store⁵
 - Comcast's feature store⁶
 - Facebook's feature store⁷
 - GO-JEK's feature store⁸
 - Twitter's feature store⁹
 - Branch International's feature store¹⁰
-
- Hopsworks' feature store¹¹ (the only open-source one!)

⁴Li Erran Li et al. "Scaling Machine Learning as a Service". In: *Proceedings of The 3rd International Conference on Predictive Applications and APIs*. Ed. by Claire Hardgrove et al. Vol. 67. Proceedings of Machine Learning Research. Microsoft NERD, Boston, USA: PMLR, 2017, pp. 14–29. URL: <http://proceedings.mlr.press/v67/li17a.html>.

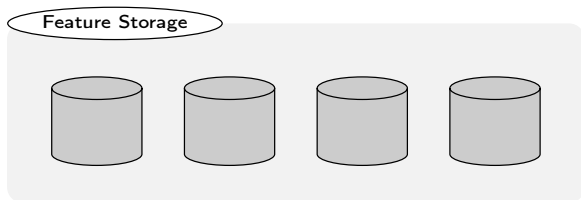
⁵Nikhil Simha and Varant Zanoan. *Zipline: Airbnb's Machine Learning Data Management Platform*. <https://databricks.com/session/zipline-airbnbs-machine-learning-data-management-platform>. 2018.

⁶Nabeel Sarwar. *Operationalizing Machine Learning—Managing Provenance from Raw Data to Predictions*. <https://databricks.com/session/operationalizing-machine-learning-managing-provenance-from-raw-data-to-predictions>. 2018.

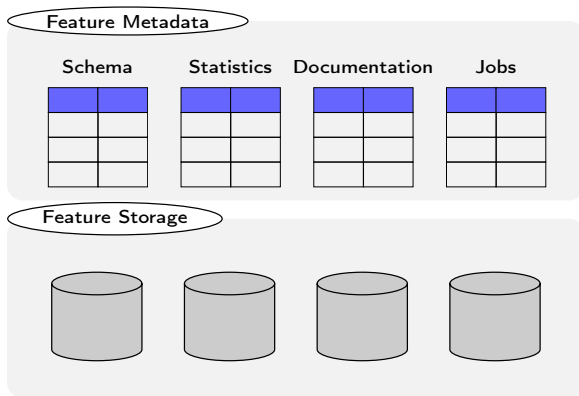
⁷Kim Hazelwood et al. "Applied Machine Learning at Facebook: A Datacenter Infrastructure Perspective". In: Feb. 2018, pp. 620–629. DOI: 10.1109/HPCA.2018.00059.

⁸Willem Pienaar. *Building a Feature Platform to Scale Machine Learning* | *DataEngConf BCN '18*. <https://www.youtube.com/watch?v=0iCXY6VnpCc>. 2018.

The Components of a Feature Store



The Components of a Feature Store



The Components of a Feature Store

Client Interface

API

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

Feature Registry



Feature Metadata

Schema

Statistics

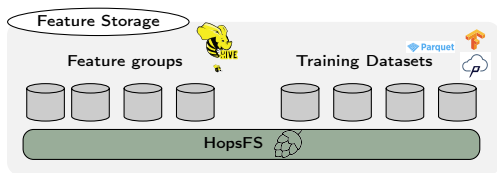
Documentation

Jobs

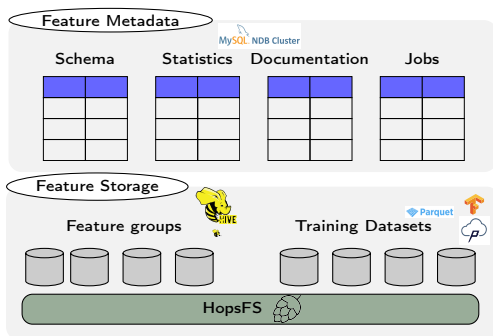
Feature Storage



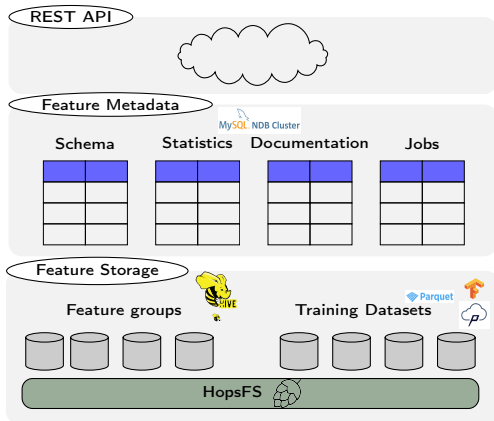
Hopworks Feature Store



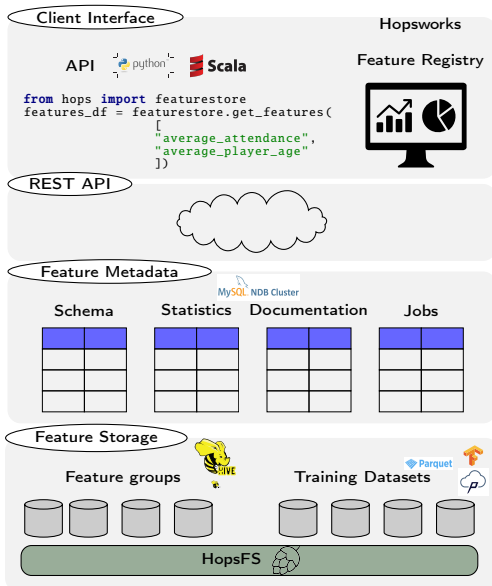
Hopworks Feature Store



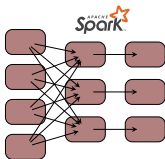
Hopworks Feature Store



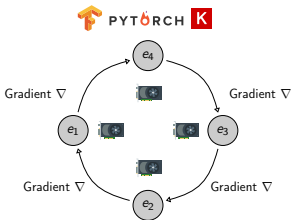
Hopsworks Feature Store



Hopsworks Feature Store




Feature Engineering



Model Training/Serving

Client Interface

API  

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

Hopsworks

Feature Registry



REST API



Feature Metadata

Schema

Statistics

Documentation

Jobs


Schema	



Statistics	

Documentation	

Jobs	

Feature Storage

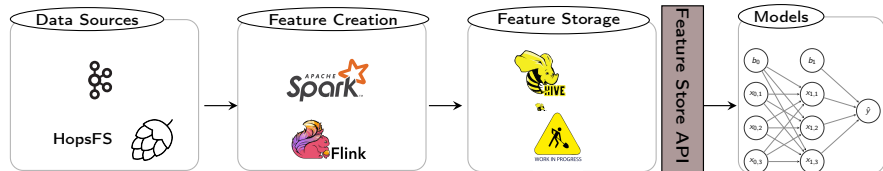
Feature groups 

Training Datasets  

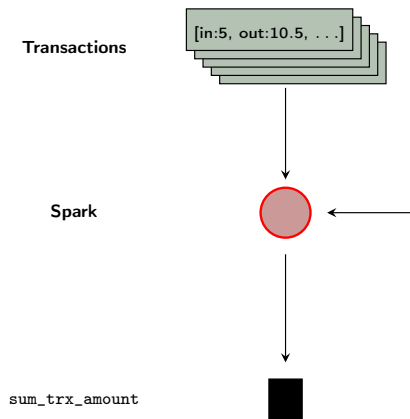


HopsFS 

Hopsworks Feature Store



Feature Creation Example



```
from hops import featurestore
trx_df = spark.read.parquet(..)
trx_sum_amount_df = trx_df.select("amount, customer")
                          .groupBy("customer")
                          .agg(sum("amount"))

featurestore.create_featuregroup(
    trx_sum_amount_df,
    "trx_sum_amount",
    description="sum of transactions"
)
```

Demo-Setting

Raw/Structured Data



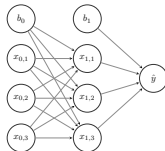
Feature
Computation



Curated Features



Model



Summary

- Machine learning comes with a high technical cost
- Machine learning pipelines needs 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*¹² provides the world's first open-source feature store



@hopshadoop
www.hops.io

@logicalclocks
www.logicalclocks.com

LOGICAL CLOCKS

We are open source:

<https://github.com/logicalclocks/hopworks>
<https://github.com/hopshadoop/hops>

13

¹²Jim Dowling. *Introducing Hopworks*. <https://www.logicalclocks.com/introducing-hopworks/>. 2018.

¹³Thanks to Logical Clocks Team: Jim Dowling, Seif Haridi, Theo Kakantousis, Fabio Buso, Gautier Berthou, Ermias Gebremeskel, Mahmoud Ismail, Salman Niazi, Antonios Kouzoupis, Robin Andersson, Alex Ormenisan, and Rasmus Toivonen

- Hopsworks' feature store¹⁴
- HopsML¹⁵
- Hopsworks¹⁶

¹⁴Kim Hammar and Jim Dowling. *Feature Store: the missing data layer in ML pipelines?* <https://www.logicalclocks.com/feature-store/>. 2018.

¹⁵Logical Clocks AB. *HopsML: Python-First ML Pipelines*. <https://hops.readthedocs.io/en/latest/hopsml/hopsML.html>. 2018.

¹⁶Jim Dowling. *Introducing Hopsworks*. <https://www.logicalclocks.com/introducing-hopsworks/> 2018.