Uddeholm Data Architecture/ML Workshop
Feature Store: the missing data layer in ML pipelines?[1]

Kim Hammar

kim@logicalclocks.com

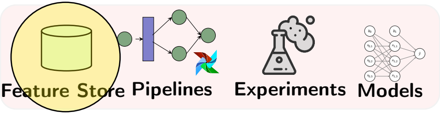
May 8, 2019

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

APIs

```python
from hops import featurestore
from hops import experiment
featurestore.get_features([
                          "average_attendance",
                          "average_player_age"])
experiment.collective_all_reduce(features, model)
```

ML/AI Assets

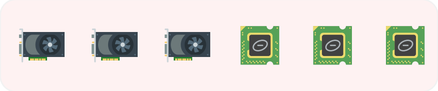


Frameworks

(ML/Data)

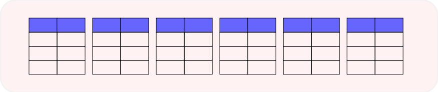
HopsYARN

(GPU/CPU as a resource)

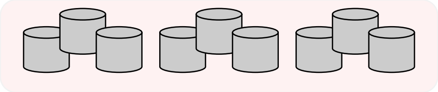


Distributed Metadata

(Available from REST API)



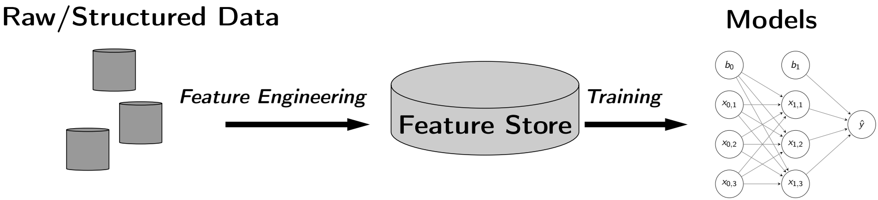
HopsFS



Outline

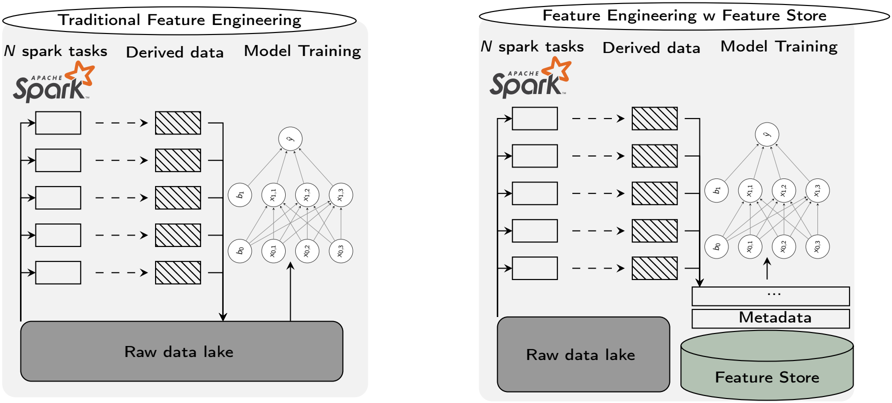
1. **What** is a Feature Store
2. **Why** You Need a Feature Store
3. **How** to Build a Feature Store (Hopsworks Feature Store)
4. Demo

Solution: Disentangle ML Pipelines with a Feature Store



- 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

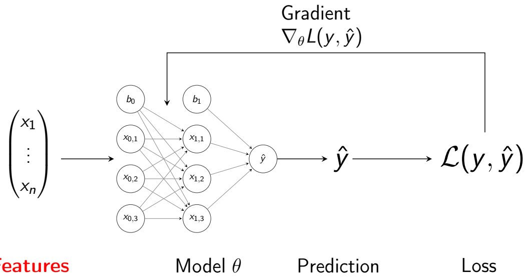
What is a Feature?

A feature is a measurable property of some data-sample

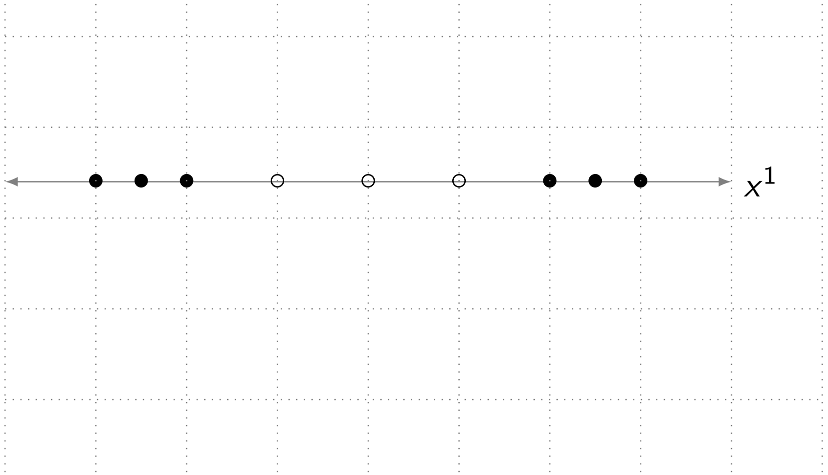
A feature could be..

- 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

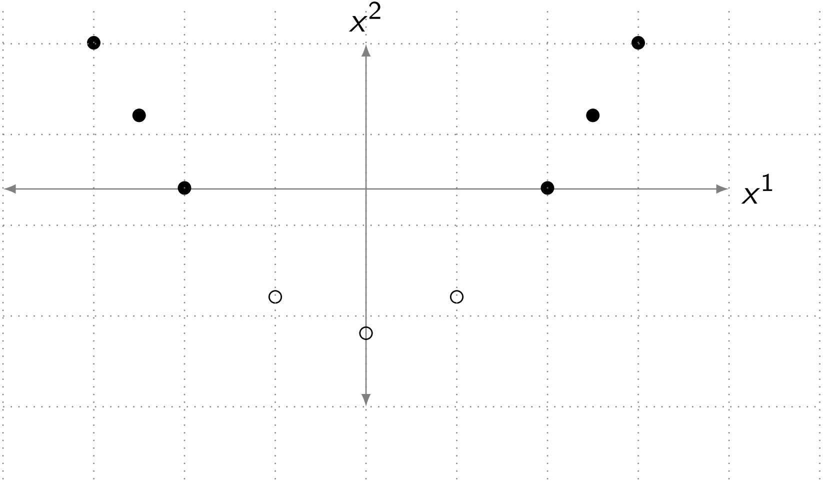
Features are the fuel for AI systems:



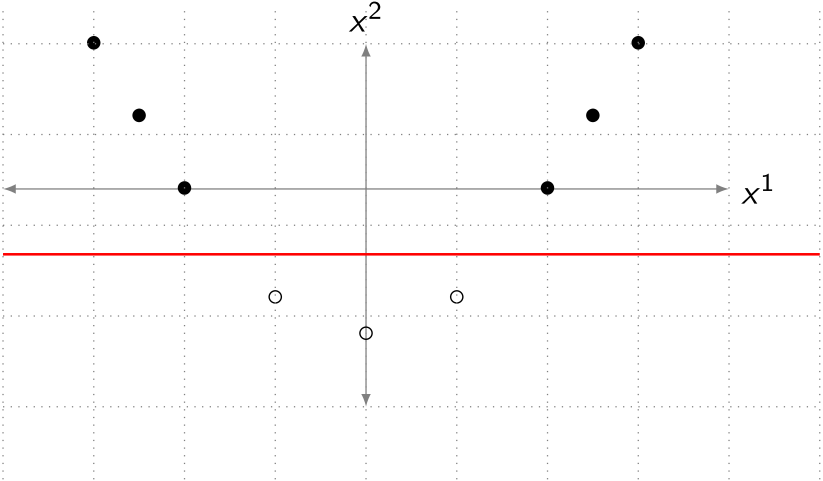
Feature Engineering is Crucial for Model Performance



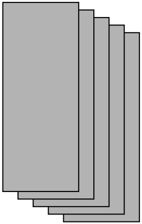
Feature Engineering is Crucial for Model Performance



Feature Engineering is Crucial for Model Performance

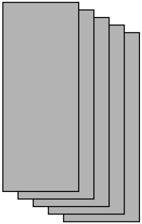


Feature Engineering is Complex



Input Data

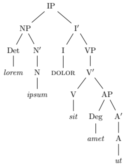
Feature Engineering is Complex



Input Data

Linguistic Features

lorem	- - ->	verb
ipsum	- - ->	noun
dolor	- - ->	stopword
sit	- - ->	noun
amet	- - ->	verb
ut	- - ->	adjective

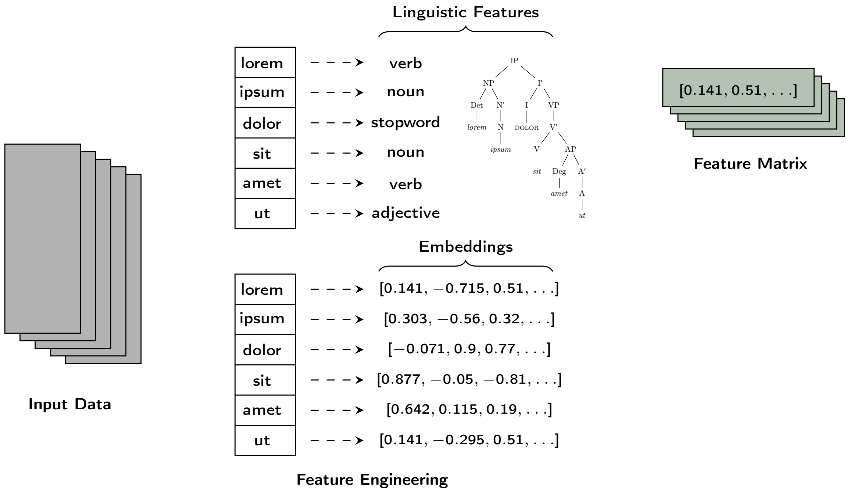


Embeddings

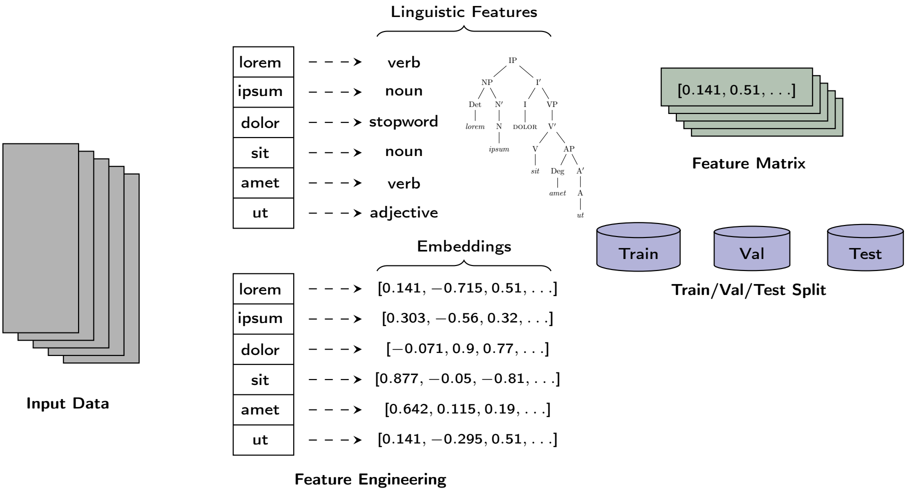
lorem	- - ->	$[0.141, -0.715, 0.51, \ldots]$
ipsum	- - ->	$[0.303, -0.56, 0.32, \ldots]$
dolor	- - ->	$[-0.071, 0.9, 0.77, \ldots]$
sit	- - ->	$[0.877, -0.05, -0.81, \ldots]$
amet	- - ->	$[0.642, 0.115, 0.19, \ldots]$
ut	- - ->	$[0.141, -0.295, 0.51, \ldots]$

Feature Engineering

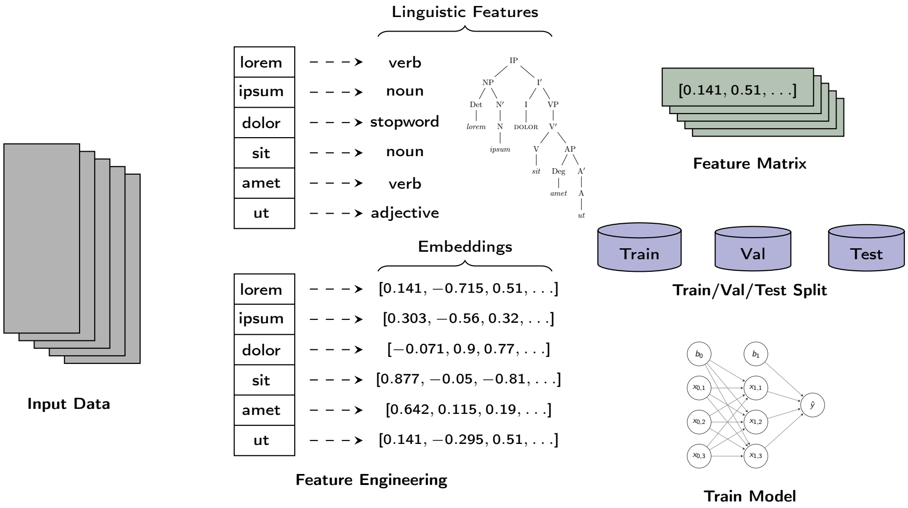
Feature Engineering is Complex



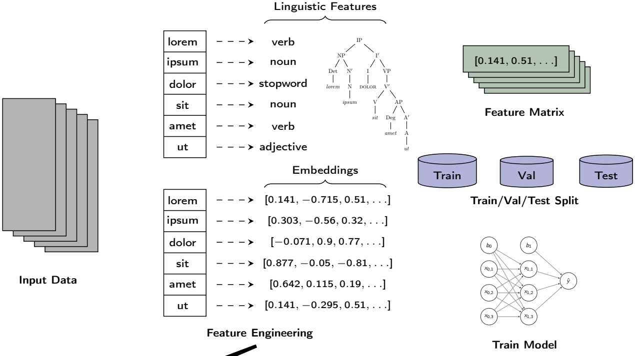
Feature Engineering is Complex



Feature Engineering is Complex

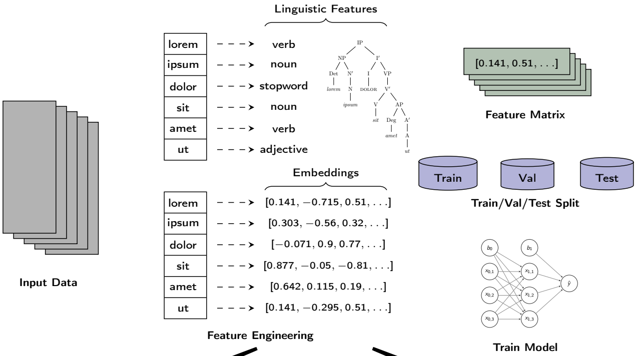


Feature Engineering is Complex



How do you make this scale?

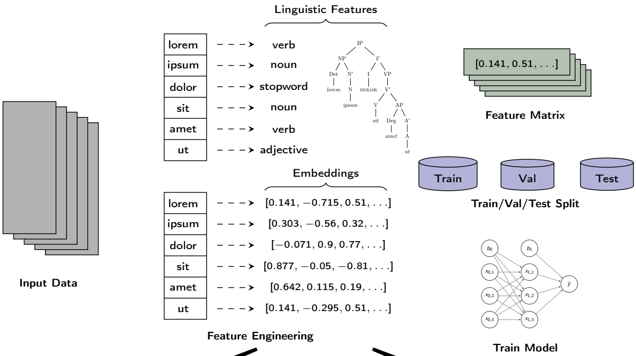
Feature Engineering is Complex



How do you make this scale?

How to manage the feature pipelines?

Feature Engineering is Complex

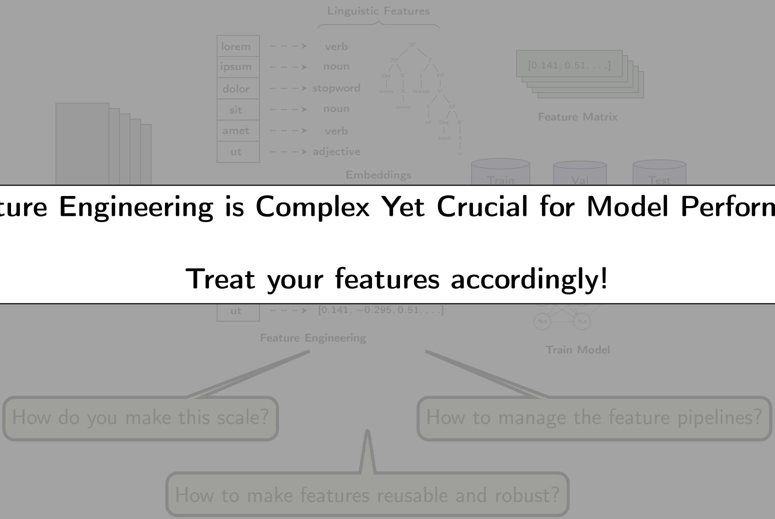


How do you make this scale?

How to manage the feature pipelines?

How to make features reusable and robust?

Feature Engineering is Complex



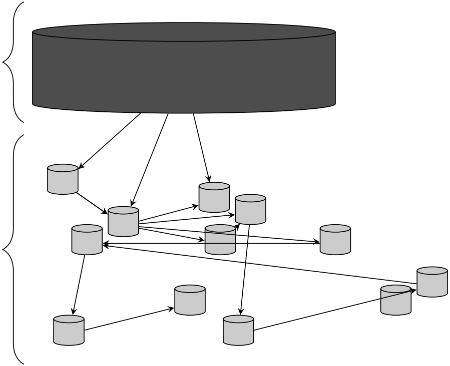
Feature Engineering is Complex Yet Crucial for Model Performance

Treat your features accordingly!

Feature Pipeline Jungles

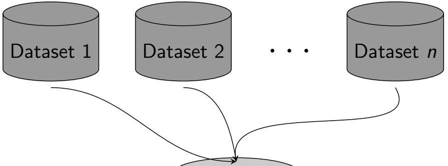
Data Lake (Raw/Structured Data)

Feature Data (Derived Data)



Disentangle Your ML Pipelines with a Feature Store

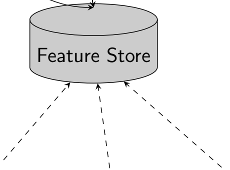
Data Sources



Feature Store

A data management platform for machine learning.

The interface between data engineering and data science.

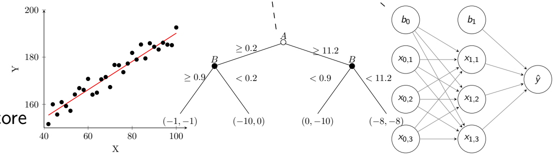


Models

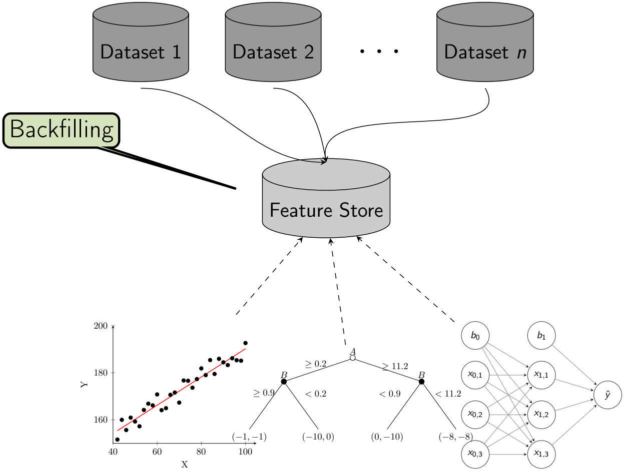
Models are trained using sets of features.

The features are fetched from the feature store

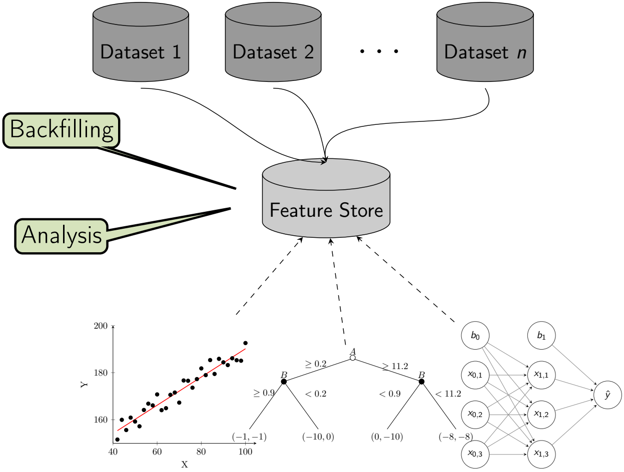
and can overlap between models.



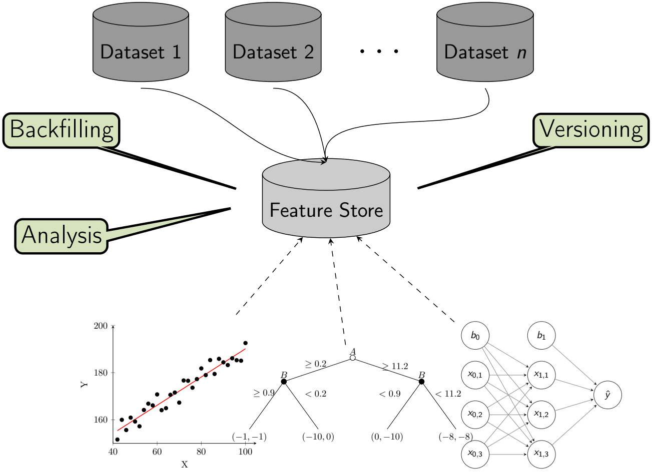
Disentangle Your ML Pipelines with a Feature Store



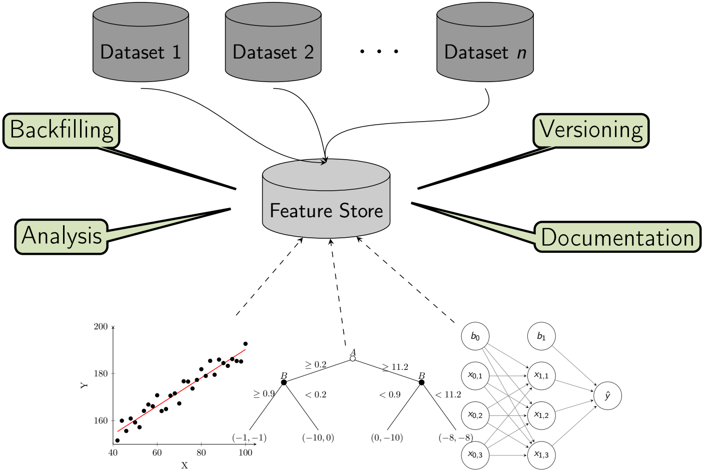
Disentangle Your ML Pipelines with a Feature Store



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 store[2]
- Airbnb's feature store[3]
- Comcast's feature store[4]
- Facebook's feature store[5]
- GO-JEK's feature store[6]
- Twitter's feature store[7]
- Branch International's feature store[8]

- Hopsworks' feature store[9] **(the only open-source one!)**

[2] Li 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.

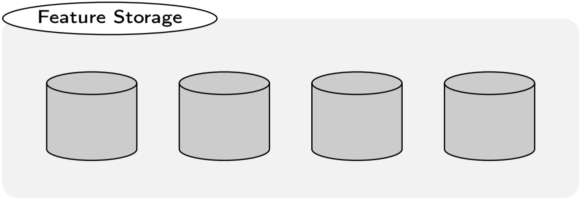
[3] Nikhil Simha and Varant Zanoyan. *Zipline: Airbnb's Machine Learning Data Management Platform*. https://databricks.com/session/zipline-airbnbs-machine-learning-data-management-platform. 2018.

[4] Nabeel 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.

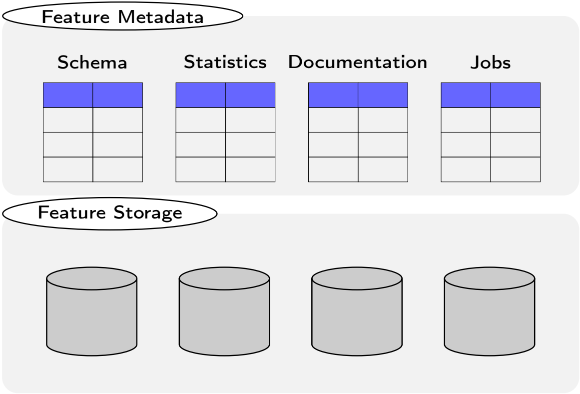
[5] Kim Hazelwood et al. "Applied Machine Learning at Facebook: A Datacenter Infrastructure Perspective". In: Feb. 2018, pp. 620–629. DOI: 10.1109/HPCA.2018.00059.

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

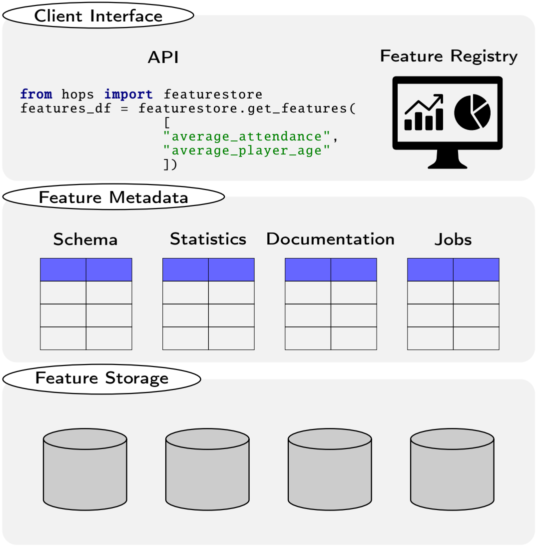
The Components of a Feature Store



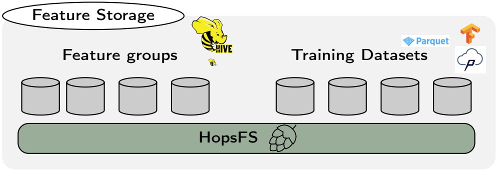
The Components of a Feature Store



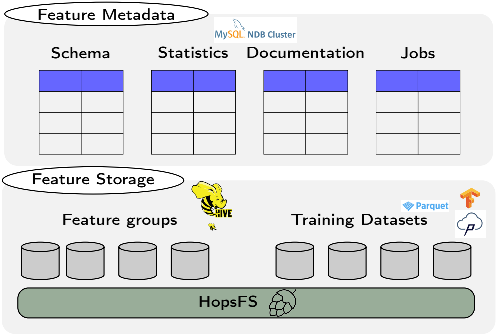
The Components of a Feature Store



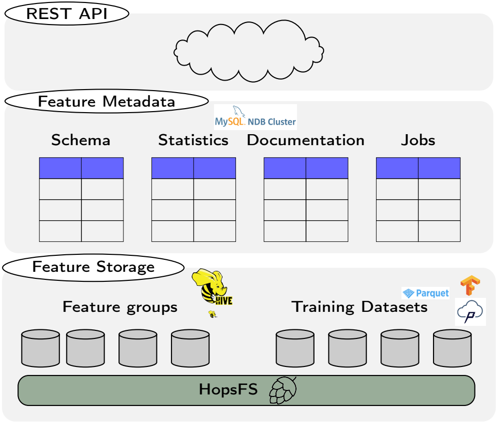
Hopsworks Feature Store



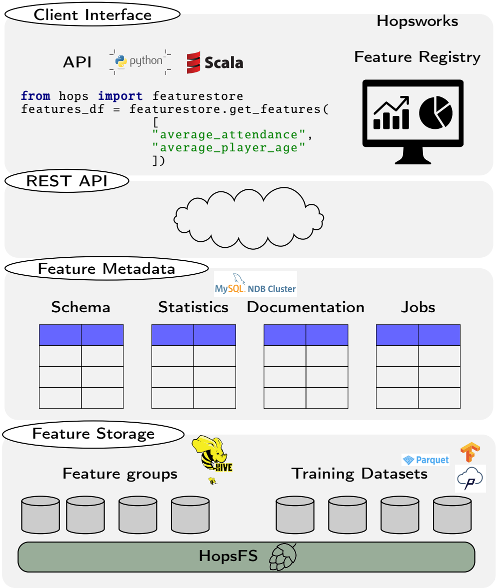
Hopsworks Feature Store



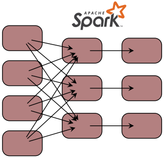
Hopsworks Feature Store



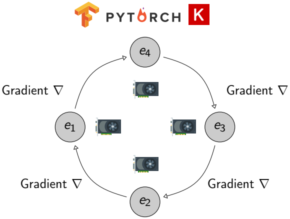
Hopsworks Feature Store



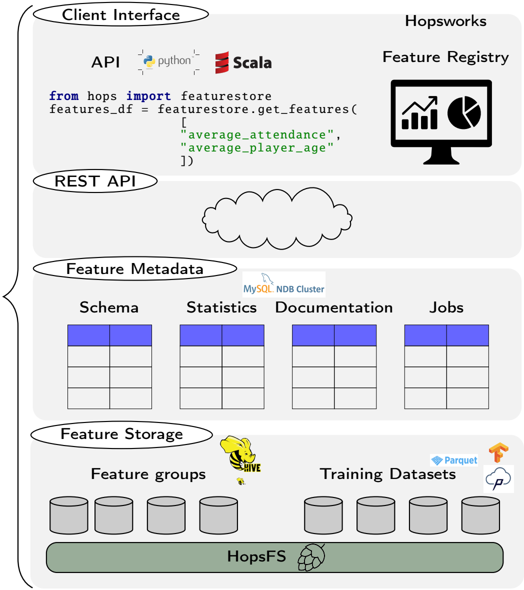
Hopsworks Feature Store



Feature Engineering



Model Training/Serving



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
- *Hopsworks*[10] 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/hopsworks
https://github.com/hopshadoop/hops

11

[10] Jim Dowling. *Introducing Hopsworks*. https://www.logicalclocks.com/introducing-hopsworks/. 2018.

[11] 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, Alex Ormenisan, Rasmus Toivonen, and Steffen Grohsschmiedt

References

- Hopsworks' feature store[12]
- HopsML[13]
- Hopsworks[14]

---

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

[13] Logical Clocks AB. *HopsML: Python-First ML Pipelines*. https://hops.readthedocs.io/en/latest/hopsml/hopsML.html. 2018.

[14] Jim Dowling. *Introducing Hopsworks*. https://www.logicalclocks.com/introducing-hopsworks/. 2018.