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

FOSDEM 2019 Brussels | HPC, Big Data and Data Science DevRoom

Kim Hammar

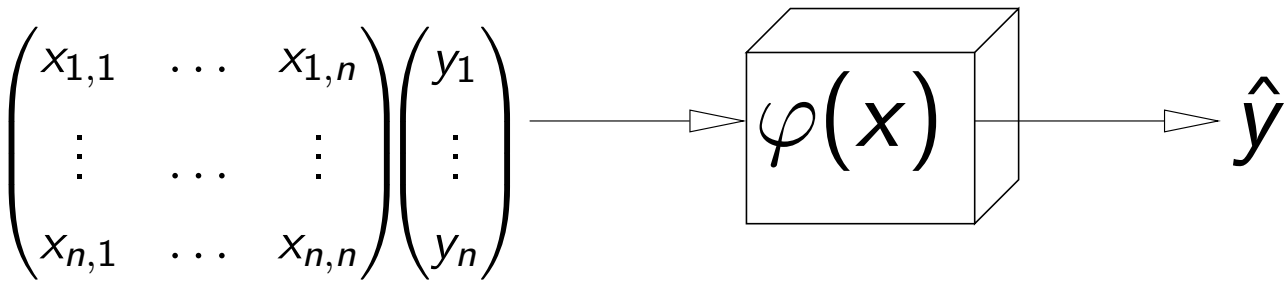
kim@logicalclocks.com

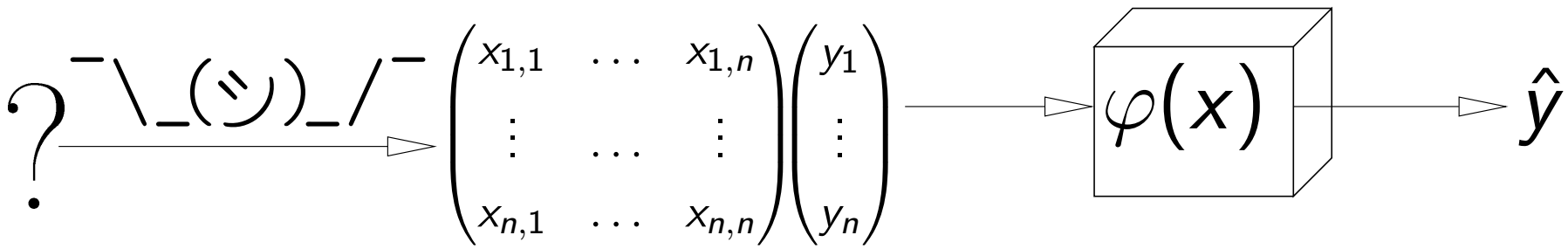
@KimHammar1

February 1, 2019

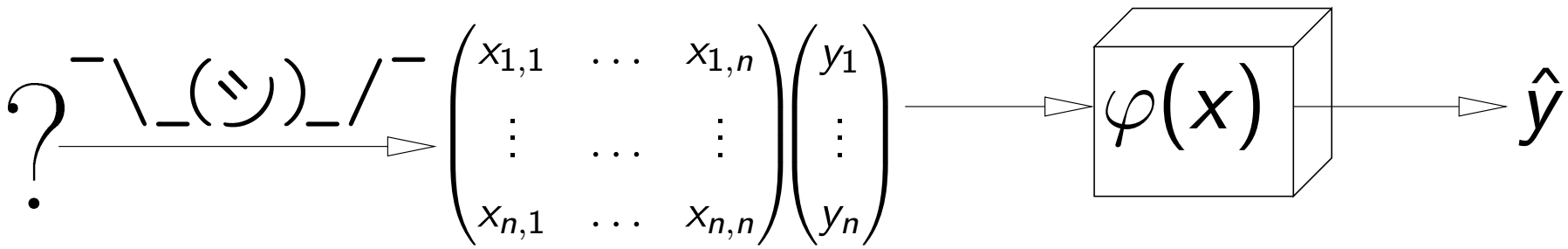
LOGICAL CLOCKS

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


$\begin{pmatrix} x_{1,1} & \cdots & x_{1,n} \\ \vdots & \cdots & \vdots \\ x_{n,1} & \cdots & x_{n,n} \end{pmatrix} \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$
$\varphi(x)$
$\hat{y}$



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

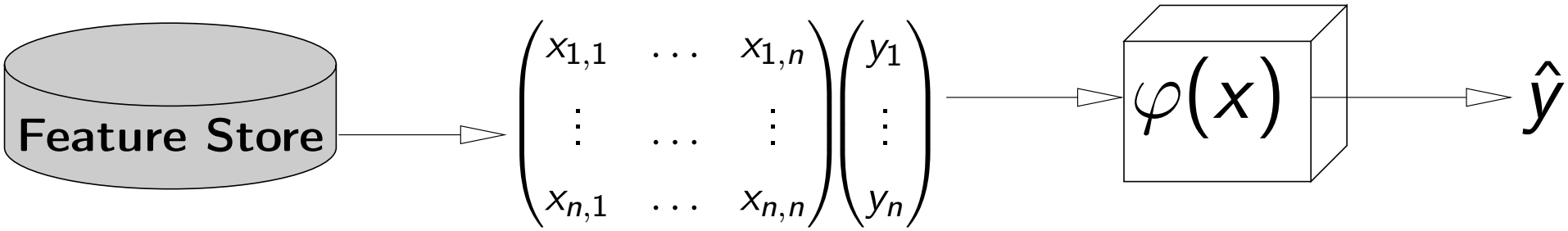


"Data is the hardest part of ML and the most important piece to get right.

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."

- Uber[2]

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



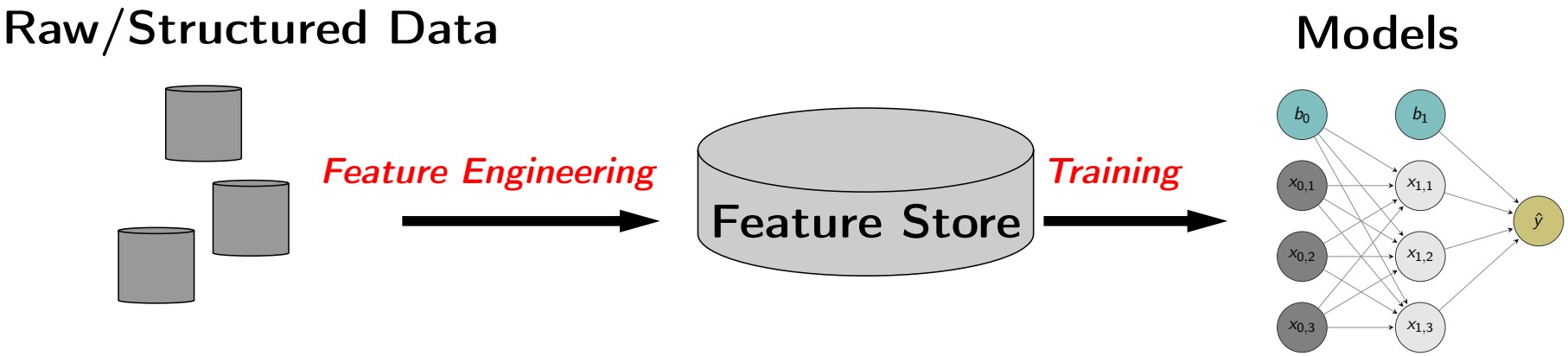
"Data is the hardest part of ML and the most important piece to get right.

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."

- Uber[3]

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

Solution: Disentangle ML Pipelines with a Feature Store

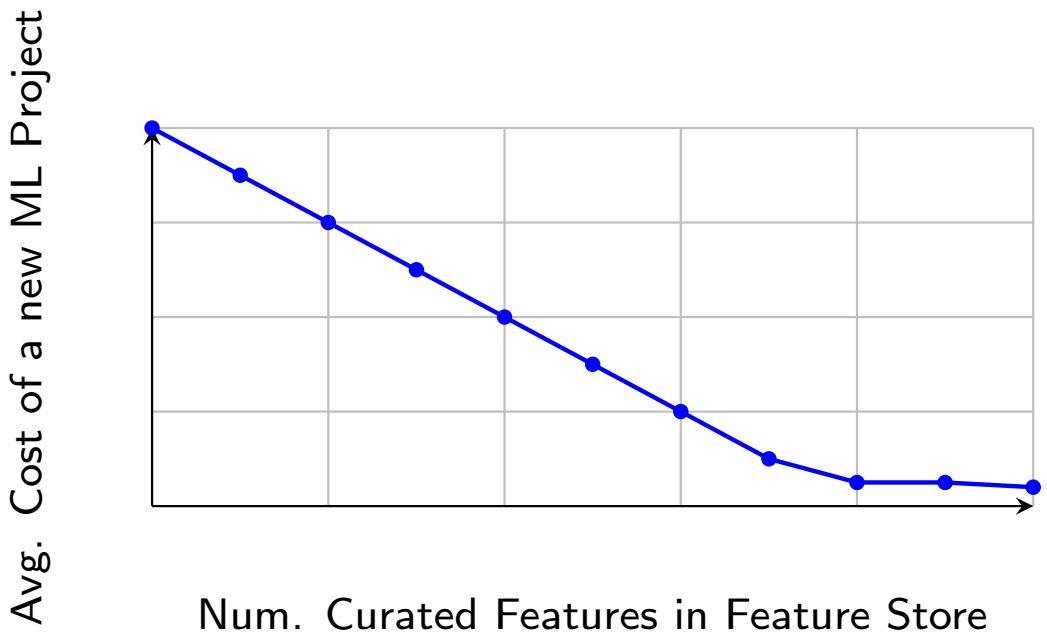


- A feature store is a central vault for storing documented, curated, and access-controlled features.
- The feature store is the interface between data engineering and data model development

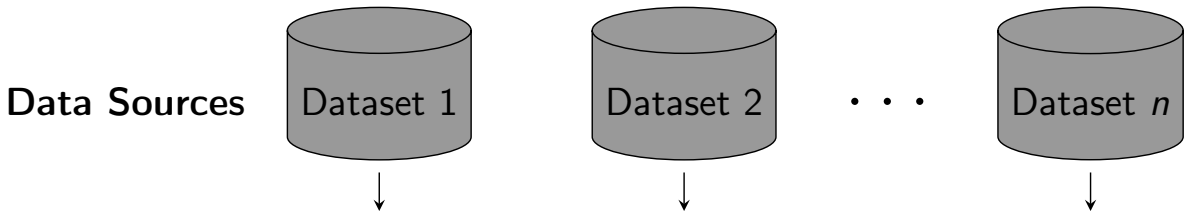
Motivation for the Feature Store

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

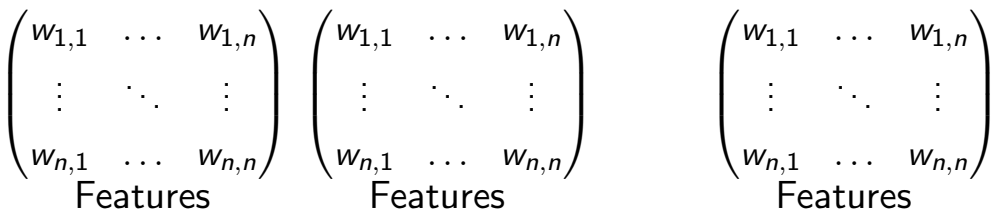


Reusing Features
Without a Feature Store is Complex



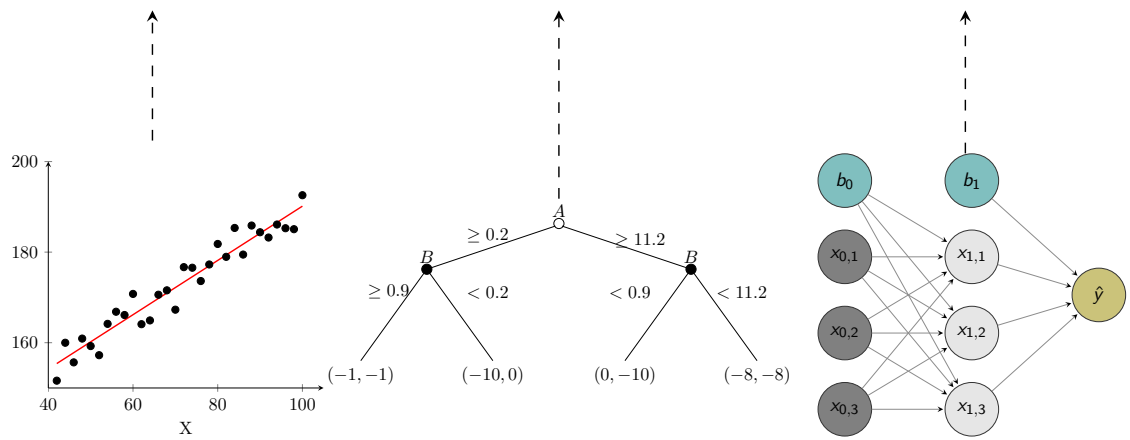
Siloed Feature Sets

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

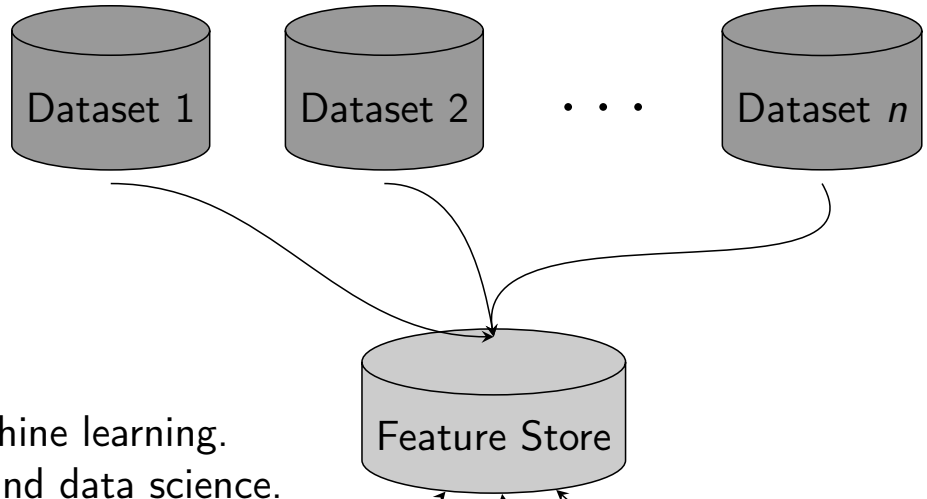


Models

Models are trained using sets of features. Without a feature store each model typically defines its own feature definitions, without feature sharing across models.

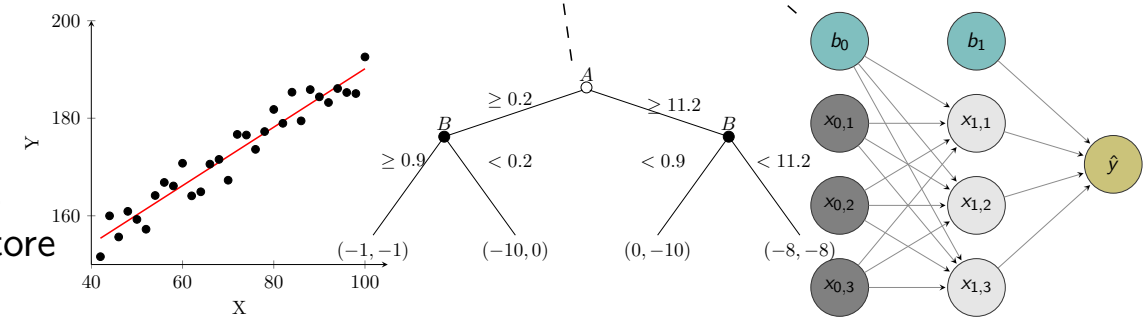


Reusing Features
With a Feature Store is Simple



Feature Store

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

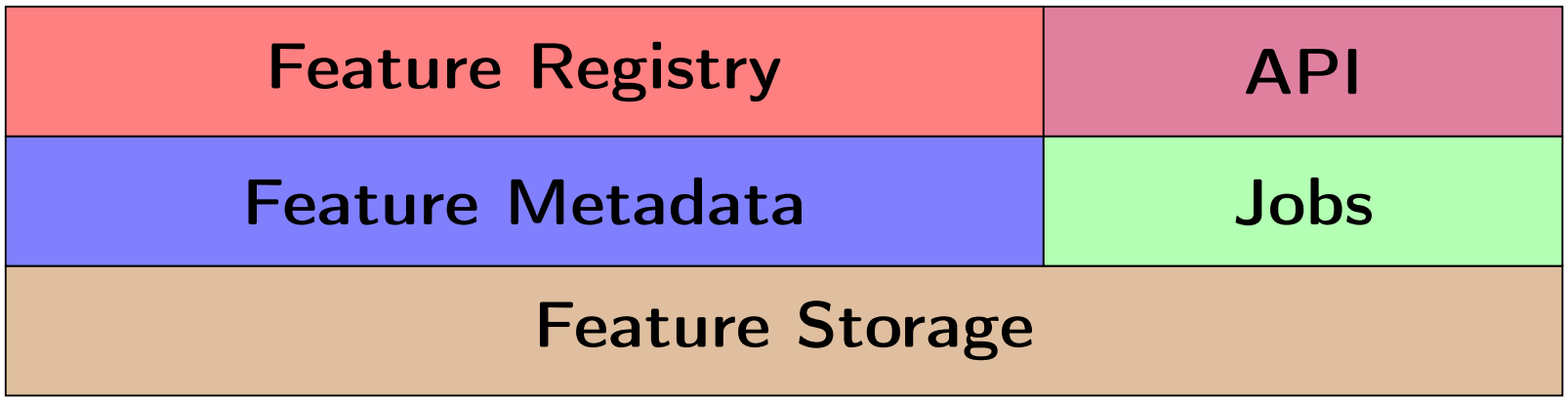


Models

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



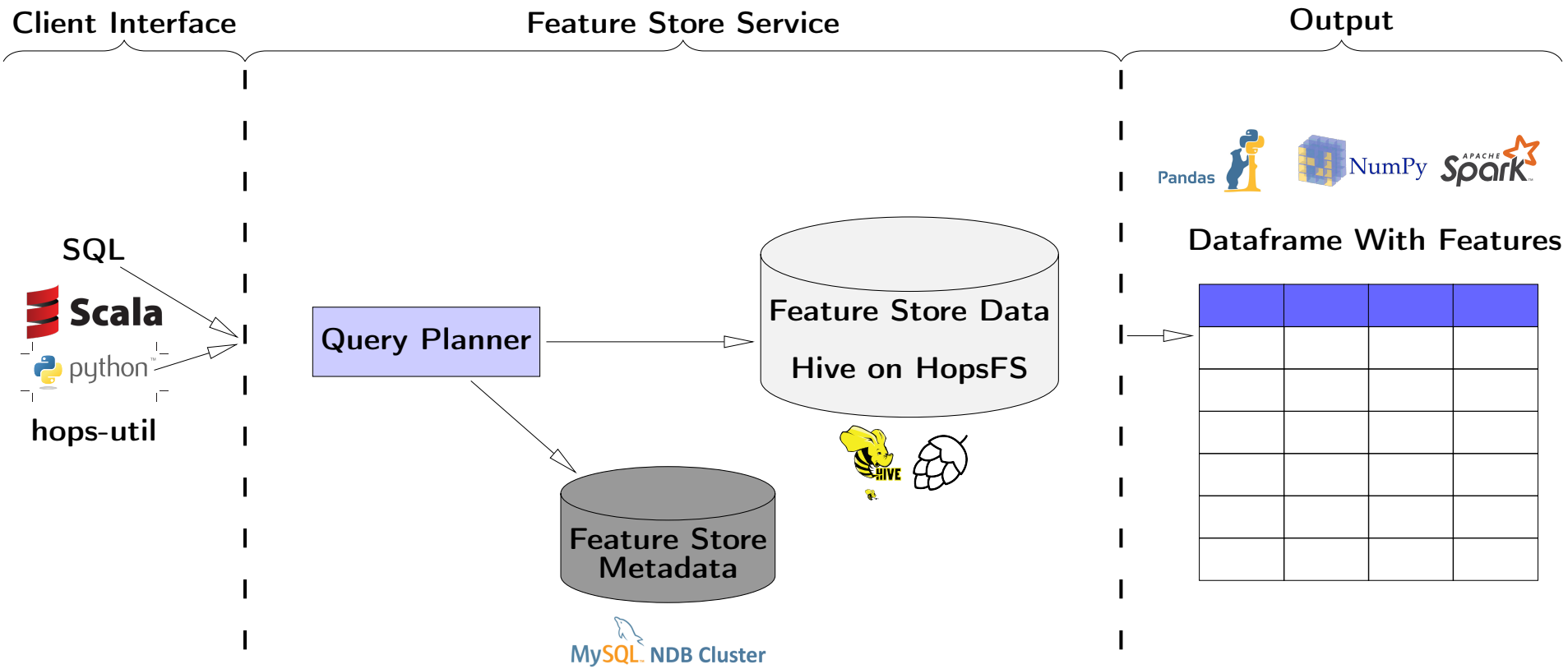
Reading from the Feature Store:

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

Writing to the Feature Store:

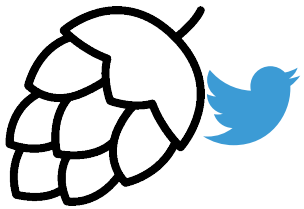
```
from hops import featurestore
raw_data = spark.read.parquet(filename)
pol_features = raw_data.map(lambda x: x^2)
featurestore.insert_into_featuregroup(pol_features, "pol_featuregroup")
```

Hopsworks Feature Store API Service



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*[4] provides the world's first open-source feature store



@hopshadoop

www.hops.io

@logicalclocks

www.logicalclocks.com

We are open source:
https://github.com/logicalclocks/hopsworks
https://github.com/hopshadoop/hops

5

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

[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[6] (**the only open-source one!**)
- Uber's feature store[7]
- Airbnb's feature store[8]
- Comcast's feature store[9]
- GO-JEK's feature store[10]
- HopsML[11]
- Hopsworks[12]

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

[7] 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.

[8] 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.

[9] 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.

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

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

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