A LOGICAL CLOCKS WHITE PAPER

HOPSWORKS FEATURE STORE

Feature Management Platform for Machine Learning

V.02

LOGICAL CLOCKS

> *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. Broken data is the most common cause of problems in production ML systems.*
>
> ***Uber on Michelangelo***

Machine learning is helping Enterprises extract knowledge from data and solve challenging prediction problems. However, machine learning is only as good as the data that it is trained on, and the first place to start when designing infrastructure for machine learning is with features.

The Feature Store is a data management system for managing machine learning features, including the feature engineering code and the feature data. The Feature Store helps ensure that features used during training and serving are consistent and that features are documented and reused within Enterprises.

• • •

The Hopsworks Feature Store enables teams to work effectively together, sharing outputs and assets at all stages in machine learning (ML) pipelines. **In effect, the Feature Store acts as an API between Data Engineering and Data Science,** enabling improved collaboration between Data Engineers, who engineer the features, with Data Scientists, who use the features to train models.

The Feature Store enables features to be registered, discovered, and used aspart of ML pipelines, thus making it easier to transform and validate the training data that is fed into machine learning systems. Hopsworks Feature Store also meets traditional Enterprise Computing requirements with support for access control, feature versioning, governance (e.g., terms of use), model interpretability, privacy, and

auditing. As machine learning models are being trained on increasingly large volumes of data, Hopsworks Feature Store is both horizontally scalable and highly available. Finally, Hopsworks Feature Store fits seamlessly into both development environments and ML pipelines – whether you are in the Cloud or on-premises.

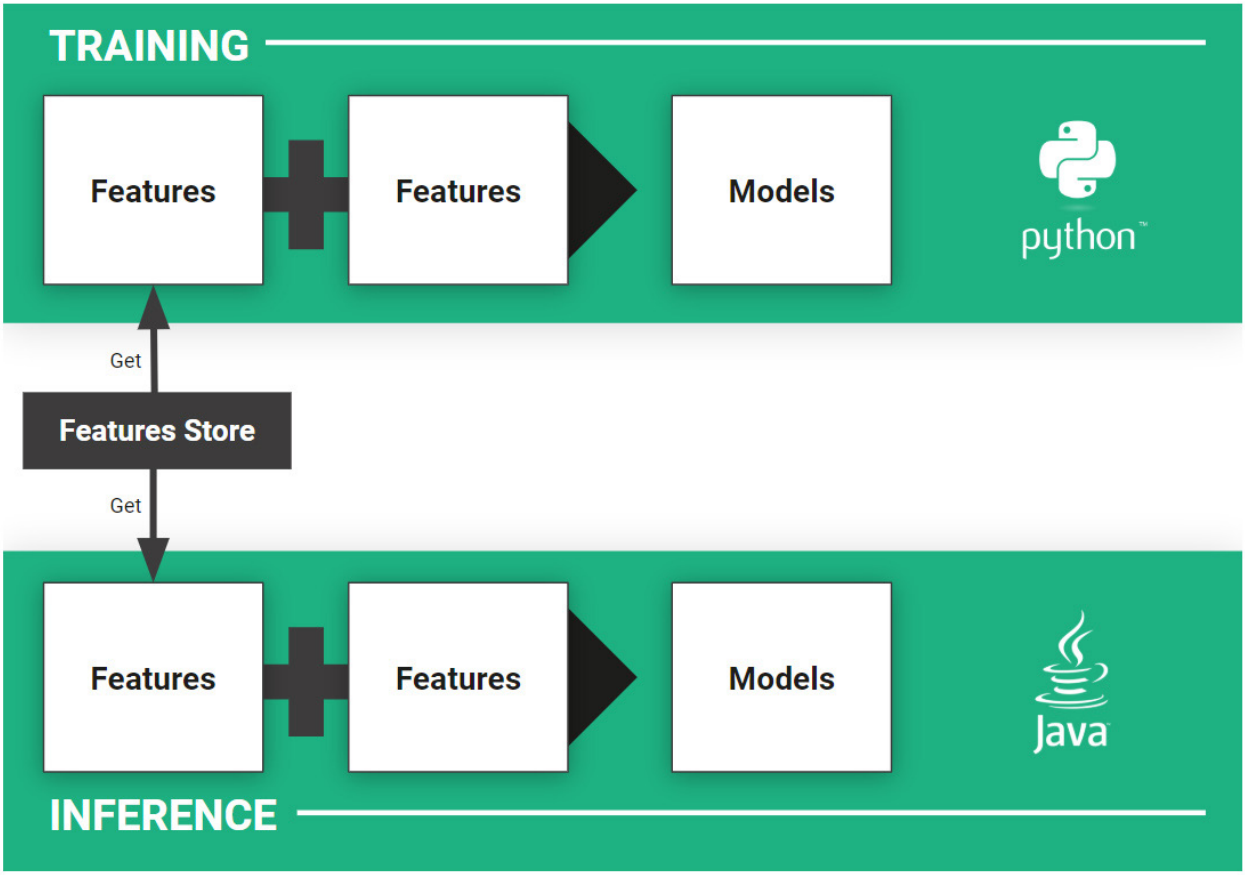


Figure 1
The Feature Store helps ensure consistent feature engineering - even when training applications arewritten in Python, while applications that use the models for inferencing are written in Java.

THE API BETWEEN DATA ENGINEERING AND DATA SCIENCE

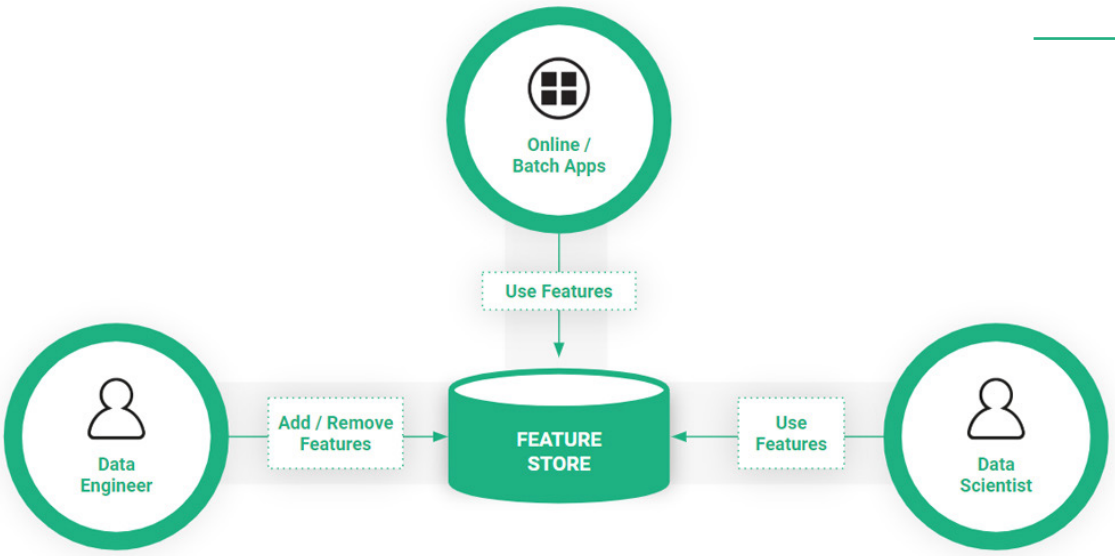


Figure 2
Align Data Engineering and Data Science Teams with the Feature Store

Feature Stores manage your Feature Data for ML, providing scalable data storage, a compute platform (to generate training datasets) and APIs to manage and use the features. The Feature Store is a key building block in building End-to-End ML workflows that take raw data, process it to produce clean feature data in the Feature Store, which is then used to create training data to produce trained ML models ready for deployment.

• • •

The Feature Store is a centralized database to store AI assets - curated features contributed by different teams throughout an Enterprise. Just as the goal of Data Warehouses/Marts/Lakes are to make all of an Enterprises analytics datasets

available for business intelligence from a single location, a Feature Store makes features for ML available in a single place, preventing feature duplication and promoting feature reuse. The Feature Store should also solve the problem of unclear feature ownership - responsibility for maintaining and update features should be clear and access to features by different users/teams should be controllable, as some features may contain sensitive data.

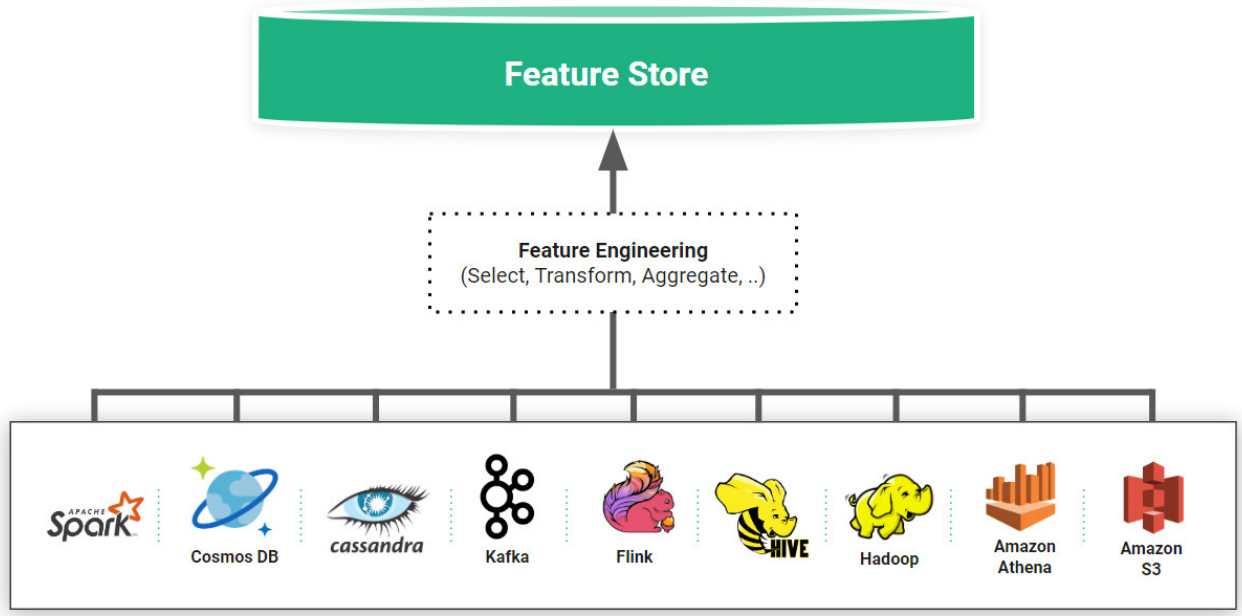


Figure 3

The Feature Store frees Data Scientists from writing Feature Pipelines that pull data from backend systems. Instead, they can now generate Train/Test data by just selecting the features of interest.

FEATURE COMPUTATION & STORAGE PLATFORM

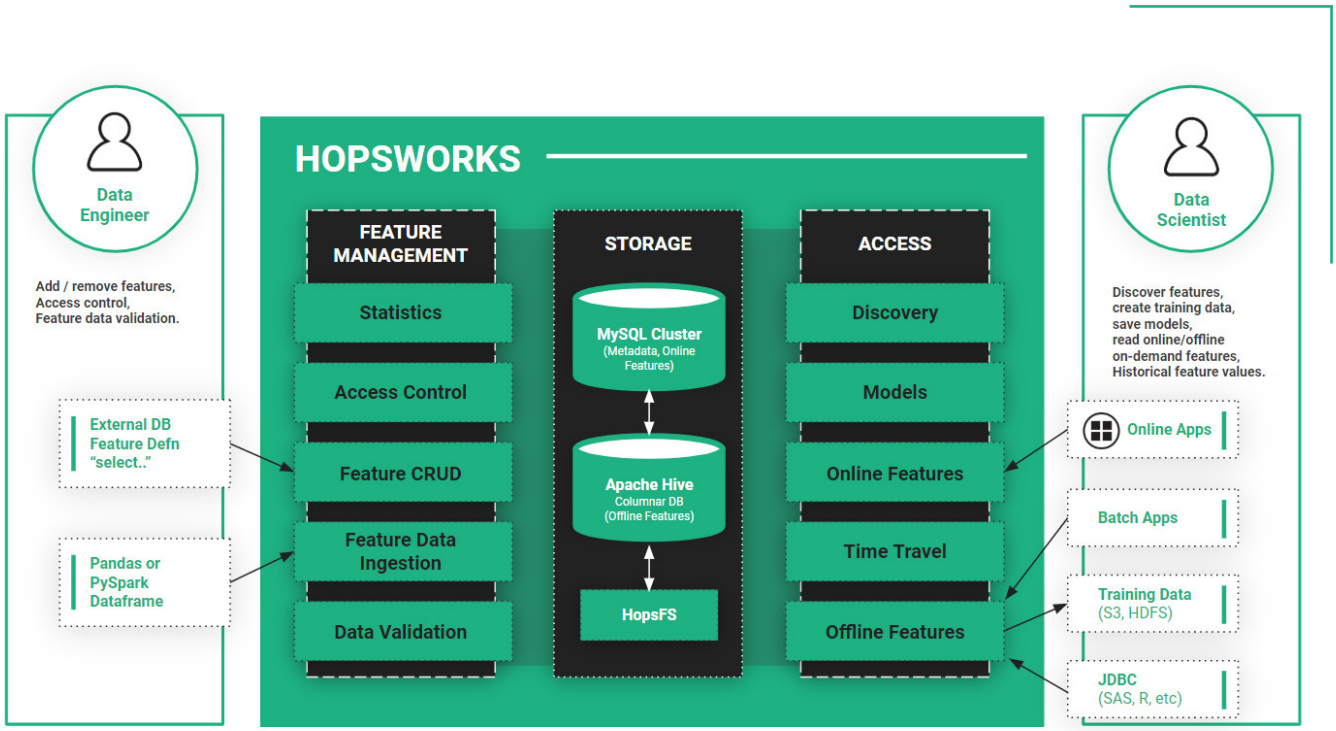


Figure 4
Hopsworks' Feature Store

In Hopsworks Feature Store, Data Engineers typically have responsibility for adding new features. **A Feature could be computed by anything from a simple SQL query on an external datastore to complex graph embeddings computed from large amounts of graph data.** Features can be created either in notebooks or programs (Python/Scala/Java APIs) or in the Hopsworks UI (for example, for on-demand features using SQL queries on external databases).

Feature data can be ingested using either a Python or Scala/Java API that takes a Pandas or Spark dataframe, and registers it as a FeatureGroup, along with user-supplied metadata for the features (name, description, etc).

The Feature data should be validated before ingestion using

our Data Validation API. Hopsworks provides both a UI-based and API-based approach for specifying data validation rules for feature data. Feature statistics can also be accessed via the Hopsworks UI or from the REST API (from, for example, a Sagemaker notebook). Just as a database management system supports the definition of more than one database, for access control support, Hopsworks supports the creation of more than one Feature Store. Additional Feature Stores are typically created to store sensitive Feature data that cannot be shared across the entire Enterprise.

Both Data Scientists and external online applications are clients of the Feature Store, with Data Scientists using it to create train/test datasets, while online applications read feature data to build feature vectors that are then sent to an online model for inferencing. In contrast to classical datawarehouses, the Feature Store also provides the ability to query historical values for features. Users can retrieve the value of features at a point in time in the past.

This is useful when a prediction outcome arrives much later, as generating a new training sample based on the outcome (label) requires joining it with the values of its features at the point of time of the prediction (a long time in the past - and the feature values may have been updated many times since then).

FEATURE STORE CONCEPTS

The Hopsworks Feature Store is a Feature Storage and Compute platform with the following properties:

- **Ingestion of ML features from Dataframes** (Spark or Pandas), Parquet files (S3, HDFS), or on-demand computation of ML features using connectors to external JDBC sources (MySQL, SQL Server, Redshift, etc);
- **Retrieval of ML features** for either (1) creating training/test datasets or (2) for serving applications that need low latency access to feature data;

- **Searchable, documented, access-controlled, versioned features**, including visualization of feature data;
- **Versioning and management of training/test datasets** in a file format of choice (.tfrecords, .csv, .numpy, etc) on a storage platform of choice (S3, HDFS, GCS, etc).

FeatureGroups, Features, & Train / Test Datasets

In Hopsworks, a Feature is an individual measurable property and each Feature belongs to a FeatureGroup, a grouping of features that are computed together share an associated key (that identifies an entitiy such as a Customer, Product, Location, Time, etc.).

Data Scientists can generate Train/Test Datasets by selecting a set of Features, a target file format, and a target storage system. The query planner will join features on a shared key and a Spark job in Hopsworks generates the Train/Test datasets in the desired file format and location.

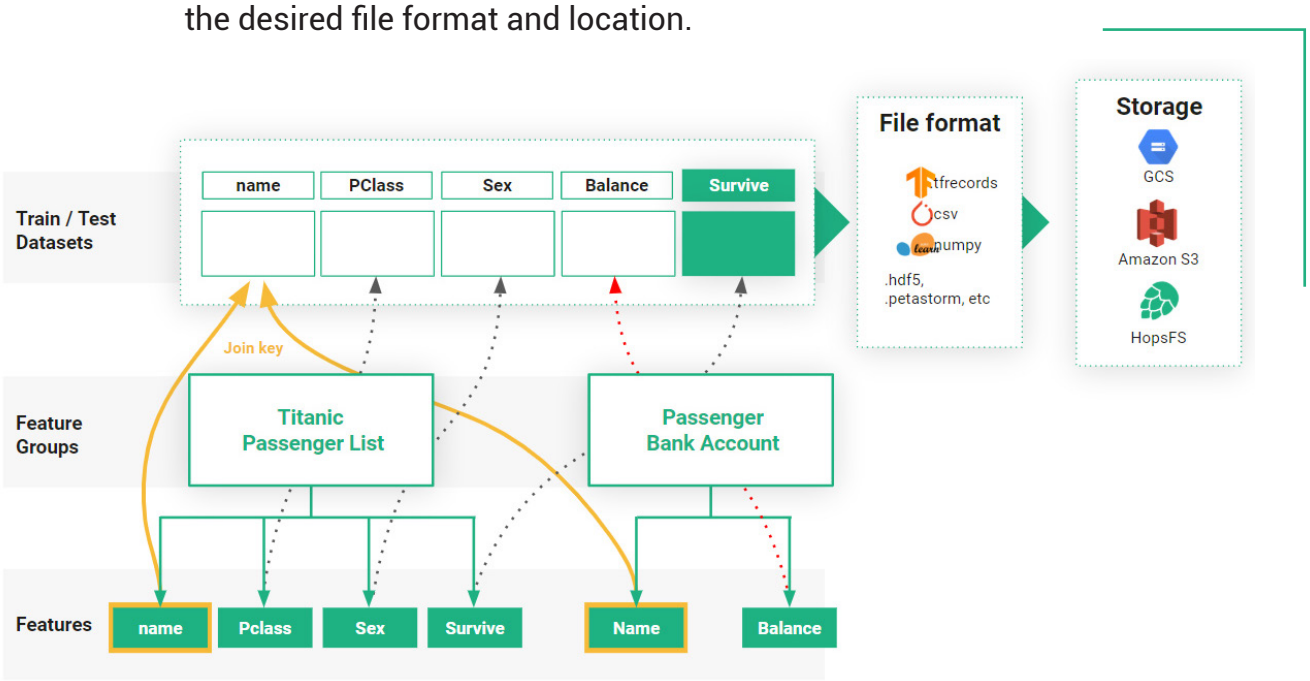
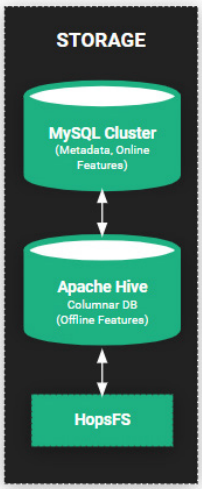


Figure 5

The Hopsworks' Feature Store consists of Features that each belong to a FeatureGroup, where the FeatureGroup has a key. Train/Test datasets can be created Hopsworks joining together Features from different FeatureGroups with a query planner (using a common key - multi-part, and usersupplied keys are also supported). The Train/Test Dataset can then be stored in a file format of choice) (TensorFlow - .tfrecords, .csv, etc) and in a storage platform of choice (e.g., S3, HopsFS, etc).

Compute On-Demand or Cache Feature Data

FeatureGroups can be either computed on-demand or cached in the Feature Store. Hopsworks supports connectors to external databases that enable Features defined on external sources (such as a SQL string on a DB2 database) to be computed on-demand. For features that are cached in the Feature Store, Hopsworks can scale to store PBs of feature data.



Sometimes features need to be pre-computed to be used by online applications (for example, a credit card fraud detection system might need to know the number of transactions executed by a customer in the last hour/day/week). For this, Hopsworks supports the online Feature Store backed by NDB (MySQL Cluster), which provides reliable low-latency feature data queries (as well as high availability within a data center and across data centers with geographical replication).

Time Travel

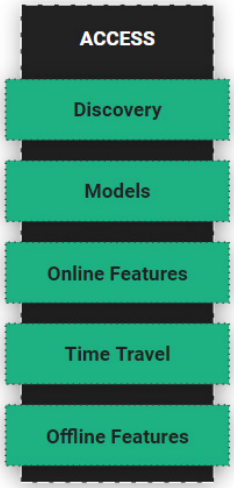
The Feature Store can also be used to help create new training data using "Time Travel" query support (backed by Apache Hudi/Hive). For example, imagine we are trying to predict credit card fraud and we get a delayed outcome (say 6 months later) that a given credit card transaction was fraudulent. We would like to add that transaction to our training dataset when we retrain the model.

However, to do so, we need to lookup the value of the features used in the prediction from 6 months ago. How may transactions did that user have in the hour/day/week before the fraudulent transaction? In a typical data warehouse that data would not be available. **However, in the Hopsworks Feature**

Store, we support time-travel queries to find out value of that feature 6 months ago.

Generate Training Data in your File Format of choice on S3/HDFS/HopsfS

The typical way-of-working for Data Scientists is to use the Feature Store to select the features that will be used to generate Train/Test data to be used to train models. Data Scientists first select the features they are interested in, then decide on what file format they want to export the training data as - .tfrecords for TensorFlow, .numpy for PyTorch, .csv, etc.



Hopsworks runs a Spark Feature Engineering job to joint the Train/Test data from all the selected features and then save the Feature data in the target file format. **The destination for the training data could be an external datalake,** such as S3/HDFS, **or the internal HopsFS filesystem in Hopsworks** (if training is to be done in the Hopsworks platform).

The Hopsworks Feature Store also provides a JDBC/ODBC API, through Apache HiveServer2, to allow cached feature data be directly accessible by external frameworks, such as SAS. Features may also need to be used for serving by online applications. Online applications typically ask for cached feature data (instead of computing features due to high latency) from Hopsworks Online Feature Store (NDB).

Batch applications, however, do not typically have latency requirments and can use either on-demand or cached features from the Offline Feature Store (Apache Hive/Hudi). Hopsworks can also manage models. Models can be published to Hopsworks or downloaded from Hopsworks using its REST API. If model training is performed in Hopsworks, **models can**

be also validated and saved/deployed using a simple python function call from the Hops python library.

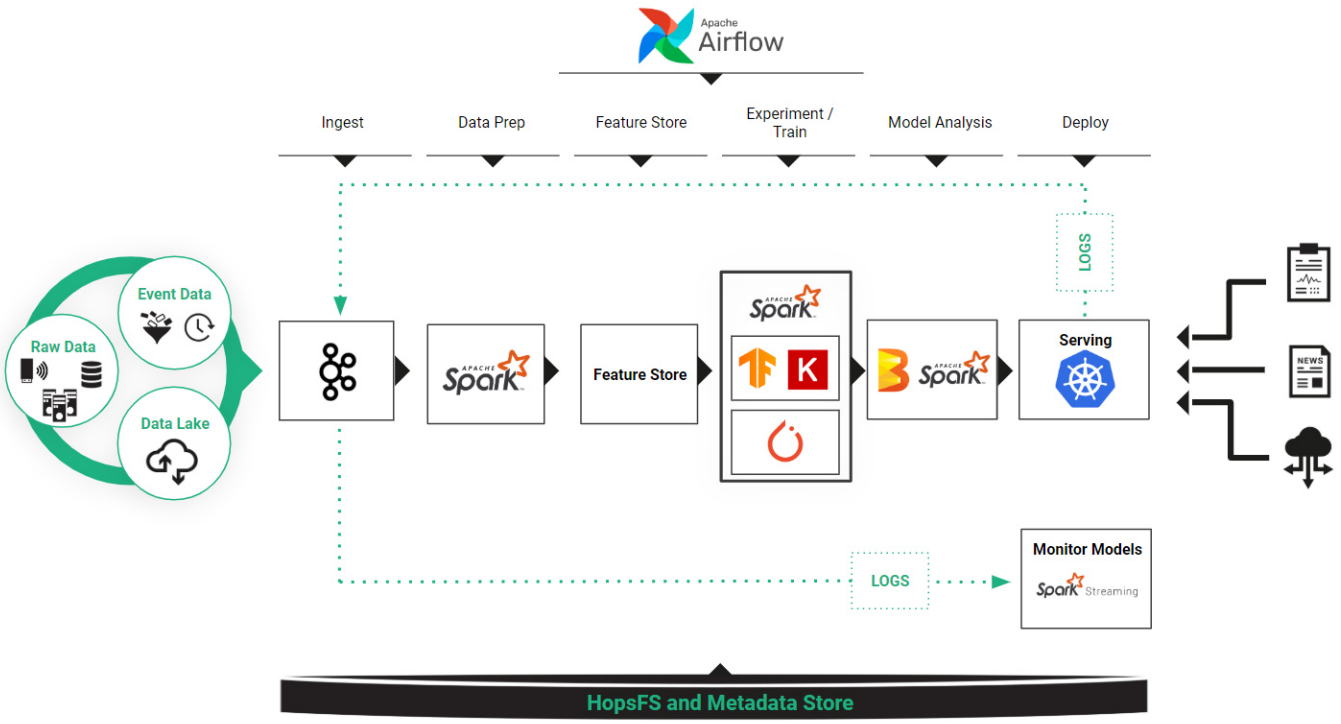


Figure 5
The Feature Store as part of an End-to-End Machine Learning Pipeline (in Hopsworks). The Feature Store can also be integrated with AWS Sagemaker, Databricks, and KubeFlow.

ML PIPELINES WITH THE FEATURE STORE

Data pipelines are challenging to develop as they are expected to be completely reliable but they have no control over their input data – it is hard to test a data pipeline against all known data inputs, when you don't know all known data inputs. ML applications differ from traditional data processing pipelines by placing additional requirements on the underlying infrastructure.

Hopsworks provides the following features to support end-toend ML pipelines using the Featue Store:

• **pipelines can start and stop at the Feature Store** - feature pipelines (that typically run when new data arrives) can run at a different cadence to training pipelines;

• **pipelines orchestrated with Apache Airflow** can launch Hopsworks Feature Engineering jobs using a Hopsworks Airflow operator;

• **pipelines run entirely on Hopsworks provide seamless integration of the Feature Store.**

UNIFIED ONLINE & OFFLINE STORAGE

The Hopsworks Feature Store consists of of dual datastores: Apache Hive a columnar database that scales to store PBs of data and and NDB (MySQL Cluster), a real-time, highly available database that can be replicated both within a data center and/or between data centers.

Hive provides access to large volumes of Feature data for creating Train/Test datasets, while NDB provide low-latency access to Feature data for serving applications. We have also extended Hive to support incremental ingestion of Feature data by integrating Hive with Hudi.

What makes Hopsworks Feature Store unique is that all of its components (Hive, HopsFS to store Hive data files, and NDB) have been adapted to use the same metadata layer, which enables us to ensure strong consistency between feature metadata and the Feature data (whether it is stored in Hive/HopsFS or NDB).

Offline Feature Data - Apache Hive and Hudi

Hive/Hudi offer a number of advantages that make it a suitable platform for storing Feature Data:

- **Open-source and scales to store PBs of Feature Data with** efficient bulk retrieval of Features;
- **LLAP support for lower latency access,** when inspecting Features from notebooks;à
- **Incremental ingestion of Feature Data** with Upsert using Apache Hudi;
- **Time-Travel queries for historical Feature** values with Apache Hudi/Hive;
- **Ability to store data files in either HopsFS or S3** (as external tables).

Online Feature Data - NDB

NDB (also known as MySQL Cluster) offers a number of advantages that make it a suitable platform for serving online Feature Data:

- **Open-source database that scales to store TBs** or Feature Data that can be queried with very low latency (~1ms);
- **Provides online applications** with real-time access to Feature Data using JDBC or native APIs;
- **Synchronous replication within a data-center** and aosynchronous replication between different NDB clusters ensures that Online Feature Data is always available;
- **NDB is a mature backend engine for MySQL** with mature tooling and operations experience.

CITIZEN DATA SCIENTISTS

Just as BI tools extended their reach to incorporate easier accessibility to both data and analytics, ML tools also need to be usable by a wider group of users in an organization. Some of the tasks needed as part of ML application development do not require knowledge of programming or statistics, but require domain knowledge of the business, knowledge of the data and its terms of use, and an ability to see relationships in the data that can have predictive insight.

The Hopsworks Feature Store provides a UI with support for:

- **search for available features;**
- **writing feature data validation rules using UI support** - to ensure valid, clean feature data;
- **data provenance to support proper** governance and (GDPR) compliance - what features were used to train which models;
- **feature store access control and creation** of more than one feature store (e.g., a company-wide feature store, and a feature store with sensitive data);
- **feature usage to provide insights into which features** are less widely used and may no longer be needed in the Feature Store;
- **feature data visualization,** to see data distributions, relationships between features, aggregate statistics on features.

USE CASE: FEATURE STORE IN BANKING

The Hopsworks Feature Store has been running in production at a Scandinavian Bank since early 2019 in an on-premises installation. As of mid 2019, the Feature Store has over 300 features created by tens of different ML projects.

Features are typically created by Data Engineers who have responsibility for a separate Hadoop-based Data Lake as well as an Enterprise Data Warehouse.

To this end, Data Engineers are given a "Data Owner" role and they have the ability to create new features. Data Scientists, however, are given a more restrictive "Data Scientist" role, where they can use the Feature Store to create new training datasets or read online feature data, but they do not have privileges to create new features.

If a Data Scientist wishes to create a new feature group, she does so in collaboration with a Data Engineer, who takes responsibility for the maintenance of the feature group.

Training using Large Datasets

One particular project required 40 TB of feature data for a single feature group. Feature Engineering was performed on the Hadoop Data Lake and engineered features were ingested via Kafka in Hopsworks to the Feature Store.

Subsequently, training datasets were created for both TensorFlow (.tfrecords files) and PyTorch (.numpy files) on HopsFS, from where models were trained using many GPUs in Hopsworks.

Integration with Active Directory (Kerberos)

The bank uses Kerberos (Active Directory / LDAP) to secure access to its existing Hadoop Data Lake. Hopsworks Feature Store was configured to support single-sign using its support for Kerberos authentication.

With a service account, Hopsworks maps existing LDAP accounts to internal users in Hopsworks. Internally, Hopsworks uses X.509 certificates (TLS) to secure data in transit and authenticate users with services.

www.logicalclocks.com
www.hopsworks.ai

Stockholm Office
Box 1263,
Isafjordsgatan 22
164 40, Kista
Sweden

Silicon Valley Office
470 Ramona St
Palo Alto,
94301 California
USA

UK Office
IDEALondon,
69 Wilson St
EC2A2BB, London
UK