An Enterprise-Grade, High Performance Feature Store - Feathr

What is Feathr?

Feathr is the feature store that is used in production in LinkedIn for many years and was open sourced in April 2022. Read our announcement on Open Sourcing Feathr and Feathr on Azure.

Feathr lets you:

- **Define features** based on raw data sources (batch and streaming) using pythonic APIs.
- **Register and get features by names** during model training and model inference.
- **Share features** across your team and company.

Feathr automatically computes your feature values and joins them to your training data, using point-in-time-correct semantics to avoid data leakage, and supports materializing and deploying your features for use online in production.

Feathr Highlights

- **Battle tested in production for more than 6 years:** LinkedIn has been using Feathr in production for over 6 years and have a dedicated team improving it.
- **Scalable with built-in optimizations:** For example, based on some internal use case, Feathr can process billions of rows and PB scale data with built-in optimizations such as bloom filters and salted joins.
- **Rich support for point-in-time joins and aggregations:** Feathr has high performant built-in operators designed for Feature Store, including time-based aggregation, sliding window joins, look-up features, all with point-in-time correctness.
- **Highly customizable user-defined functions (UDFs)** with native PySpark and Spark SQL support to lower the learning curve for data scientists.

- **Pythonic APIs** to access everything with low learning curve; Integrated with model building so data scientists can be productive from day one.
- **Derived Features** which is a unique capability across all the feature store solutions. This encourage feature consumers to build features on existing features and encouraging feature reuse.
- **Rich type system** including support for embeddings for advanced machine learning/deep learning scenarios. One of the common use cases is to build embeddings for customer profiles, and those embeddings can be reused across an organization in all the machine learning applications.
- **Native cloud integration** with simplified and scalable architecture, which is illustrated in the next section.
- **Feature sharing and reuse made easy:** Feathr has built-in feature registry so that features can be easily shared across different teams and boost team productivity.

☁️ Running Feathr on Cloud with a few simple steps

Feathr has native integrations with Databricks and Azure Synapse:

Follow the Feathr ARM deployment guide to run Feathr on Azure. This allows you to quickly get started with automated deployment using Azure Resource Manager template.

If you want to set up everything manually, you can checkout the Feathr CLI deployment guide to run Feathr on Azure. This allows you to understand what is going on and set up one resource at a time.

- Please read the Quick Start Guide for Feathr on Databricks to run Feathr with Databricks.
- Please read the Quick Start Guide for Feathr on Azure Synapse to run Feathr with Azure Synapse.

📓 Documentation

- For more details on Feathr, read our documentation.
- For Python API references, read the Python API Reference.
- For technical talks on Feathr, see the slides here. The recording is here.

🛠️ Install Feathr Client Locally

If you want to install Feathr client in a python environment, use this:

```
pip install feathr
```

Or use the latest code from GitHub:

```
pip install git+https://github.com/linkedin/feathr.git#subdirectory=feathr_project
```

Feathr Highlighted Capabilities

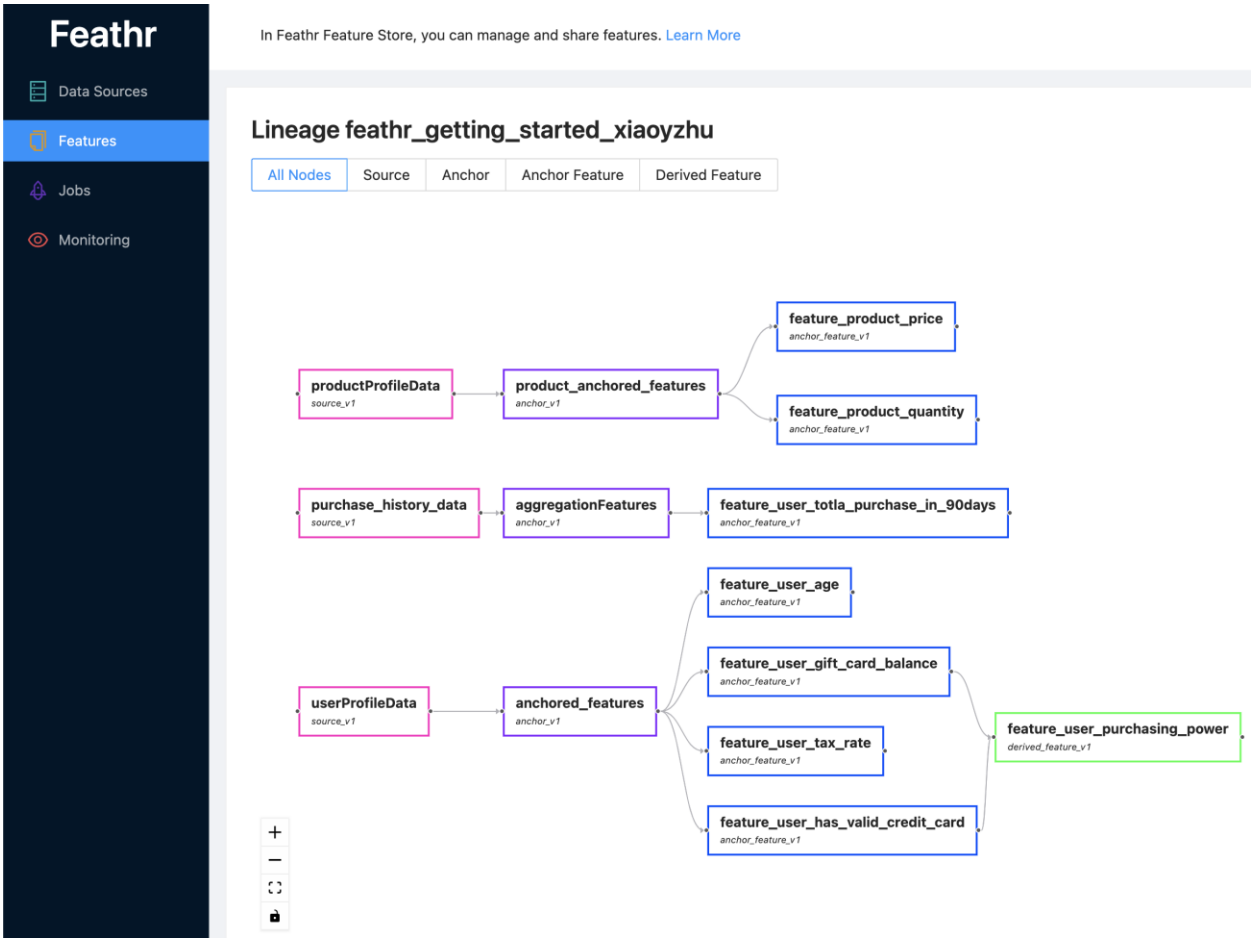
Please read Feathr Full Capabilities for more examples. Below are a few selected ones:

Feathr UI

Feathr provides an intuitive UI so you can search and explore all the available features and their corresponding lineages.

You can use Feathr UI to search features, identify data sources, track feature lineages and manage access controls. Check out the latest live demo here to see what Feathr UI can do for you. Use one of following accounts when you are prompted to login:

- A work or school organization account, includes Office 365 subscribers.
- Microsoft personal account, this means an account can access to Skype, Outlook.com, OneDrive, and Xbox LIVE.



For more information on the Feathr UI and the registry behind it, please refer to [Feathr Feature Registry]

Rich UDF Support

Feathr has highly customizable UDFs with native PySpark and Spark SQL integration to lower learning curve for data scientists:

```
def add_new_dropoff_and_fare_amount_column(df: DataFrame):
    df = df.withColumn("f_day_of_week", dayofweek("lpep_dropoff_datetime"))
    df = df.withColumn("fare_amount_cents", df.fare_amount.cast('double') * 100)
    return df

batch_source = HdfsSource(name="nycTaxiBatchSource",

path="abfss://feathrazuretest3fs@feathrazuretest3storage.dfs.core.windows.net/demo_data/green_tripdata_2020-04.csv",
                          preprocessing=add_new_dropoff_and_fare_amount_column,
                          event_timestamp_column="new_lpep_dropoff_datetime",
                          timestamp_format="yyyy-MM-dd HH:mm:ss")
```

Defining Window Aggregation Features with Point-in-time correctness

```
agg_features = [Feature(name="f_location_avg_fare",
                        key=location_id,                          # Query/join key of the
feature(group)
                        feature_type=FLOAT,
                        transform=WindowAggTransformation(        # Window Aggregation
transformation
                            agg_expr="cast_float(fare_amount)",
                            agg_func="AVG",                       # Apply average aggregation
over the window
                            window="90d")),                       # Over a 90-day window
                ]

agg_anchor = FeatureAnchor(name="aggregationFeatures",
                           source=batch_source,
                           features=agg_features)
```

Define features on top of other features - Derived Features

```
# Compute a new feature(a.k.a. derived feature) on top of an existing feature
derived_feature = DerivedFeature(name="f_trip_time_distance",
                                 feature_type=FLOAT,
                                 key=trip_key,
                                 input_features=[f_trip_distance, f_trip_time_duration],
                                 transform="f_trip_distance * f_trip_time_duration")

# Another example to compute embedding similarity
user_embedding = Feature(name="user_embedding", feature_type=DENSE_VECTOR, key=user_key)
item_embedding = Feature(name="item_embedding", feature_type=DENSE_VECTOR, key=item_key)

user_item_similarity = DerivedFeature(name="user_item_similarity",
                                      feature_type=FLOAT,
                                      key=[user_key, item_key],
                                      input_features=[user_embedding, item_embedding],
                                      transform="cosine_similarity(user_embedding,
item_embedding)")
```

Define Streaming Features

Read the Streaming Source Ingestion Guide for more details.

Point in Time Joins

Read Point-in-time Correctness and Point-in-time Join in Feathr for more details.

Running Feathr Examples

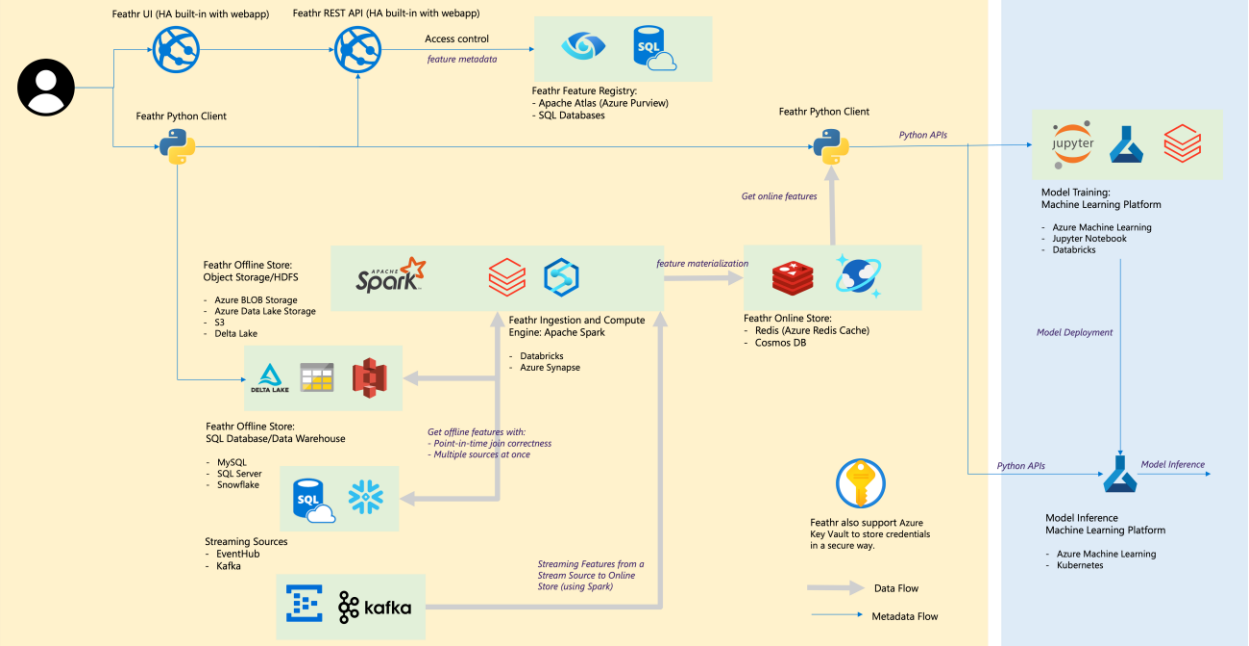
Follow the quick start Jupyter Notebook to try it out. There is also a companion quick start guide containing a bit more explanation on the notebook.

🗣 Tech Talks on Feathr

- Introduction to Feathr - Beginner's guide
- Document Intelligence using Azure Feature Store (Feathr) and SynapseML

- [Notebook tutorial: Build a Product Recommendation Machine Learning Model with Feathr Feature Store]

⚙ Cloud Integrations and Architecture



Feathr component	Cloud Integrations
Offline store – Object Store	Azure Blob Storage, Azure ADLS Gen2, AWS S3
Offline store – SQL	Azure SQL DB, Azure Synapse Dedicated SQL Pools, Azure SQL in VM, Snowflake
Streaming Source	Kafka, EventHub
Online store	Redis, Azure Cosmos DB (coming soon), Aerospike (coming soon)
Feature Registry and Governance	Azure Purview, ANSI SQL such as Azure SQL Server
Compute Engine	Azure Synapse Spark Pools, Databricks
Machine Learning Platform	Azure Machine Learning, Jupyter Notebook, Databricks Notebook
File Format	Parquet, ORC, Avro, JSON, Delta Lake, CSV
Credentials	Azure Key Vault

🚀 Roadmap

For a complete roadmap with estimated dates, please visit this page.

- ☑ Support streaming
- ☑ Support common data sources
- ☑ Support feature store UI, including Lineage and Search functionalities

- ❑ Support online transformation
- ❑ Support feature versioning
- ❑ Support feature monitoring
- ❑ Support feature data deletion and retention

👨‍👩‍👧‍👦 Community Guidelines

Build for the community and build by the community. Check out Community Guidelines.

📢 Slack Channel

Join our Slack channel for questions and discussions (or click the invitation link).