FEATURE STORES & THEIR ROLE IN MLOPS

Feature stores are a part of the machine learning workflow and ML tooling. Feature stores fit themselves between the data and the machine learning models.

What is a feature store?

Feature stores are a way to manage data, particularly for machine learning operations (MLOps).

A simple explanation of MLOps is all the engineering pieces you have to bring together in order to deploy, run, and train AI models. As such, MLOps weaves together significant components of:

- Machine learning
- DevOps
- Data engineering

With feature stores, the craftsmanship of ML teams can increase because their data—the lifeblood of any ML model—is more easily seen.

(Read our machine learning vs AI comparison.)

How feature stores work

Data used to sit comfortably in a server somewhere. Data scientists would access the data when it was needed for analysis. Or, users would query the server to display data.

But, in the era of big data, this type of data storage and recall is no longer feasible.

Machine learning, and the construction of AI systems, presents a different kind of use case: Different data needs to be pulled from their sources, and then transformed and combined to create a standard feature set to serve a model.

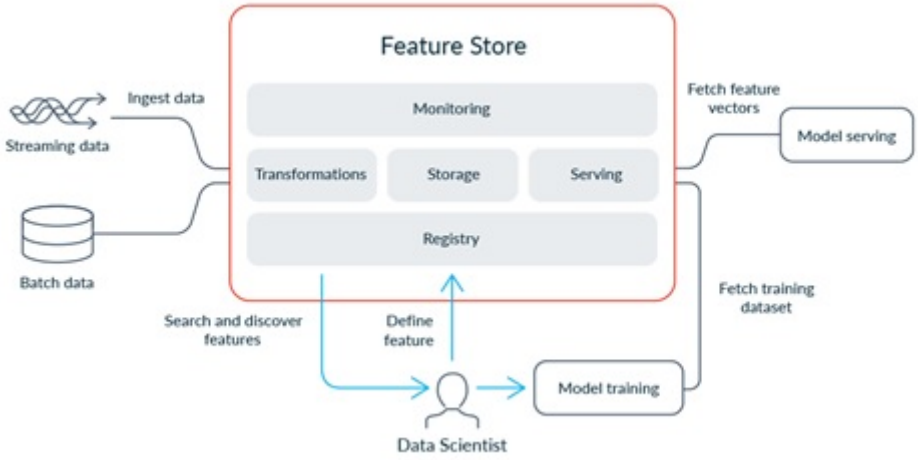
Teams face challenges when putting ML models into production. Mike Del Basco, the co-founder of Tecton.ai and a creator of Uber's Michelangelo ML platform, says teams face these challenges:

- Accessing the right raw data
- Building features from raw data
- Combining features into training data
- Calculating and serving features in production
- Monitoring features in production

Feature stores solve a problem around the increasing complexity of data demands made by the ML workflow. Data scientists want—need—to be able to easily:

- Verify the validity of the data
- Check its quality
- Know it's fresh
- Version the data and share it with others

Feature stores are positioned between the ML models and the data sources to accommodate the kind of accessibility that Machine Learning teams require.



(Source)

Feature stores help teams:

- Better collaborate
- Reduce duplication
- Better regulatory compliance
- Faster development

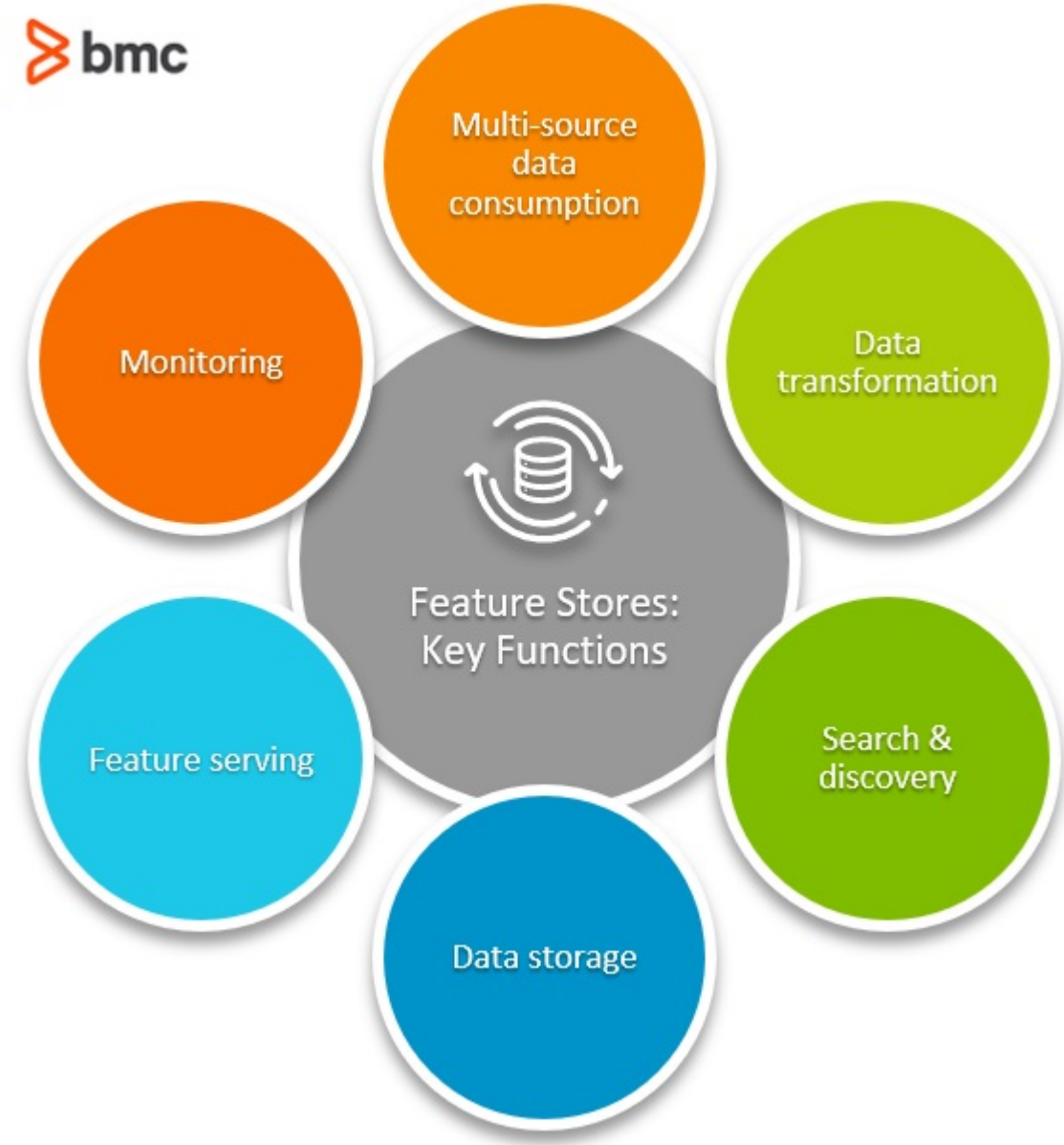
Where data is the fundamental component of an ML model, feature stores increase a team's ability to:

1. Communicate about their data
2. Create more unique ML models.

Key functions of feature stores

Feast is an open source feature store project built in collaboration with Google and Gojek. Its Google deployment is accessed through an API. Feast is built upon Redis, then uses Big Query for storage.

Today, Feast operates on every major platform. Still, its design practice is a good model for what to look for in a feature store. So, let's explore key functions and components of feature stores.



Multi-source data consumption

Features stores will load data from upstream sources. Data comes from many sources. Features will need to be extracted from those sources and combined into feature sets.

The feature store needs to be able to read from many sources. A feature store can load data through:

- Various streams
- Warehouses
- Data files

Data transformation

A key benefit of the feature store is it makes it easy to use different types of features together in the same models.

Data scientists will pull data from different sources, and the feature store allows the model to join or transform data in consistent ways, while monitoring the action was completed, and completed successfully.

User	*Last Active*	*Date Joined*
id57890	3/14/2021	1/1/2019
id57314	3/12/2021	1/8/2019

Data table #1

# Entries	Sentiment Score	Role
45	0.45	SVP
13	0.89	VP

Data table #2

User	# Entries	Sentiment Score
id57890	45	0.45
id57314	13	0.89

Data table #3

Along with creating feature sets from different sources, data gets transformed in a number of ways to support a model. One way is one-hot encoding or labeling data, or encoding text data to be processed by a model.

The feature store will verify there is consistency between these transformations with proper analytics and monitoring to make ensure the data has successfully completed the step before being served to a model.

Without the verification step, it's possible for you to train models and serve them into production—all on stale and incomplete data.

Search & discovery

A major design consideration with the feature store is to encourage collaboration among teams. When one feature set is created and works well with a model, the aim is for that set to be shared and consumed by other models or created by other teams.

A good feature store will have ways of sharing sets, with registry information to ensure the features are standardized and made consistent across all teams' models.

Data storage

Feature stores need to store data somewhere, either online or offline, as data from the sources passes through it.

From Tecton.ai:

- **"Offline storage layers are typically used to store months' or years' worth of feature data for training purposes.** Offline feature store data is often stored in data warehouses or data lakes like S3, BigQuery, Snowflake, Redshift. Extending an existing data lake or data warehouse for offline feature storage is typically preferred to prevent data silos.
- **Online storage layers are used to persist feature values for low-latency lookup during inference.** They typically only store the latest feature values for each entity, essentially modeling the current state of the world. Online stores are usually eventually consistent, and do not have strict consistency requirements for most ML use cases. They are usually implemented with key-value stores like DynamoDB, Redis, or Cassandra."

Feature serving

The feature store needs to be able to serve features to different models, and the tooling to do this is generally through an API.

Models also require consistency across the features served to it, so when the features are served, a check can verify that the data being served fits the requirements the model demands.

Monitoring

Finally, the always present concern around blocks of code is integrating the ability to monitor the feature store. The feature store should provide a lot of metrics on the data and can discover the completeness, correctness, and quality of datasets travelling through it.

The monitoring helps the system to be updated, debugged in events of error, and advanced in complexity.

Getting started with feature stores

When ML teams get big, finding a way to maintain their data becomes a necessity. Most of the big companies that utilize AI have already built internal feature stores: Google, Uber, Facebook, Amazon, Salesforce, etc.

Now those tools are becoming a MLaaS for newer companies who are advancing down their AI journey.

Here are the tools for developing your own feature store:

- Tecton.ai
- Feast.ai (Open source feature store)
- Amazon Sagemaker
- Molecula
- Feast/Kubeflow integration

Growing MLOps has more needs

Feature stores are just one of the emerging necessities of large ML Ops teams. The MLOps discipline has high demands at higher operational levels. For true MLOps, you need to be able to:

- Make data access and data quality easy.
- Follow the development of a model from its beginning through training and into serving.
- Monitor the success of the model when it is served into production.

In general, we see the ML industry grouping around three key areas:

- Data management, including data annotation and verification, quality control, and data accessibility to models and data scientists
- Data pipelines that automate the ML workflow
- Serving, or putting the model into production

As we can see, feature stores fit nicely into these groups.

Related reading

- BMC Machine Learning & Big Data Blog
- Data Architecture Explained: Components, Standards & Changing Architectures
- Data Annotation & Its Role in Machine Learning
- Top Machine Learning Frameworks To Use
- 3 Simple Data Strategies for Companies