Accelerating ML at Uber with Michelangelo Palette

Amit Nene, Tech Lead Manager

Uber Technologies Inc.

FEATURE STORE SUMMIT 2022

Organized by HOPSWORKS

Uber Engineering

Michelangelo Palette

Evolution into a Feature Engineering Platform

Shared repository

Offline Store

Online Store

Batch feature computation

Near real-time feature computation

Scalability

Reliability

Feature Quality & Monitoring

Automatic Feature Selection

Feature APIs

Provenance

Embeddings

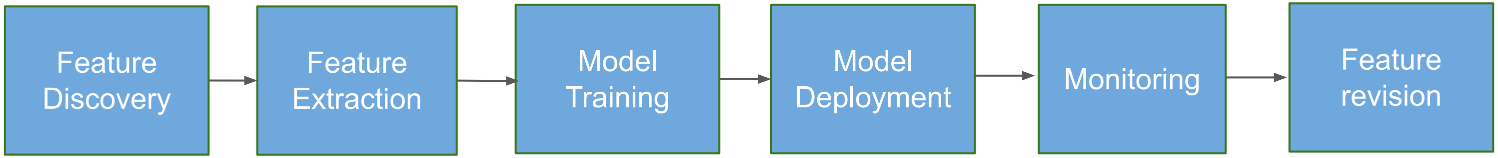
2016 → 2022

Feature Engineering: User journey

Number of trips taken over last N days

Number of orders from a particular restaurant

Average rating of user



Feature Discovery
- Discover relevant features

Feature Extraction
- Build new features or embeddings
- Production pipelines: Batch, NRT

Model Training
- Feature Preparation at scale
- Feature Quality

Model Deployment
- Online serving at scale

Monitoring
- Monitor Model
- Online/Offline Feature drift
- Online and Offline Feature Quality

Feature revision
- Deprecate pipelines
- Share features

FEATURE STORE SUMMIT 2022

Feature Discovery

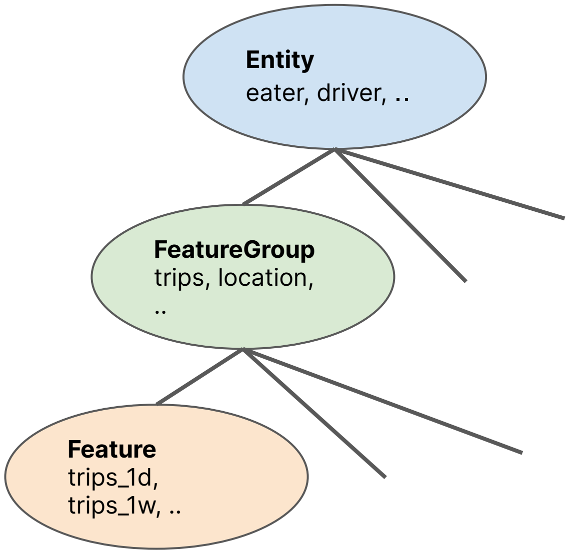
Browsing through Git repo

Entity: Primary entity associated with a Feature Group (table)

Feature Group: A logical table of individual features

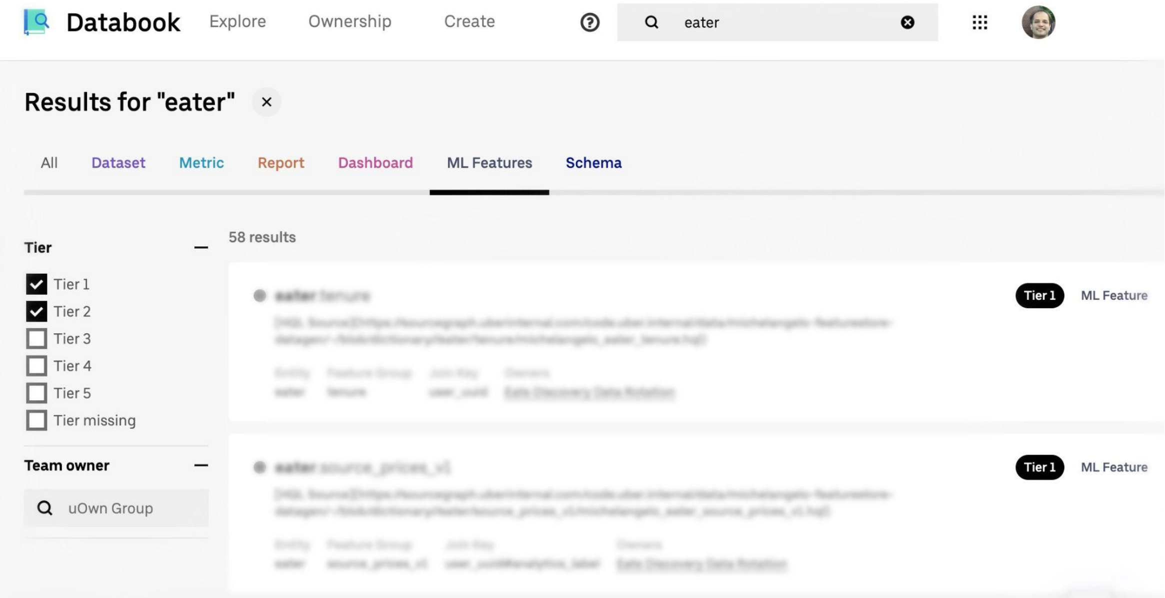
Feature: A logical column within a Feature Group

Metadata: Type information, Join/Lookup Key, Feature extraction ETL, Online dispersal info



Feature Discovery

Keyword Search



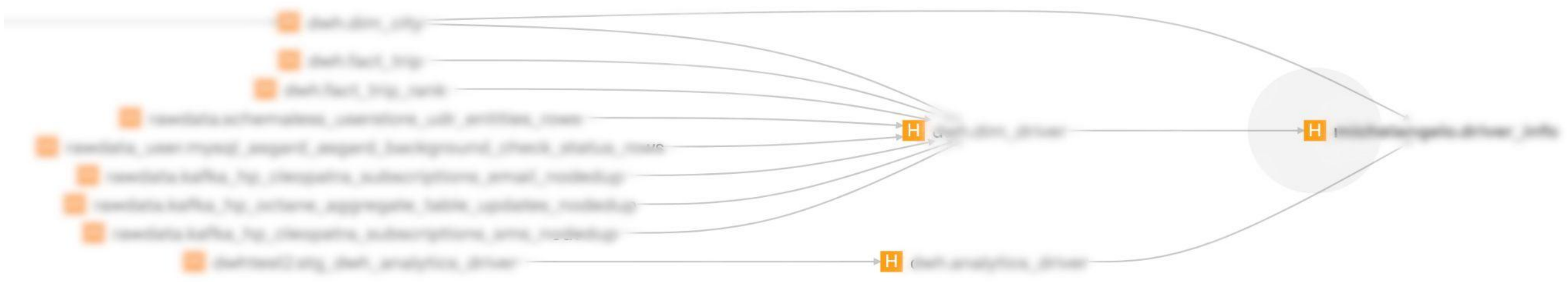
Feature Discovery

Data Lineage

Warehouse: dim, fact, business tables, event logs

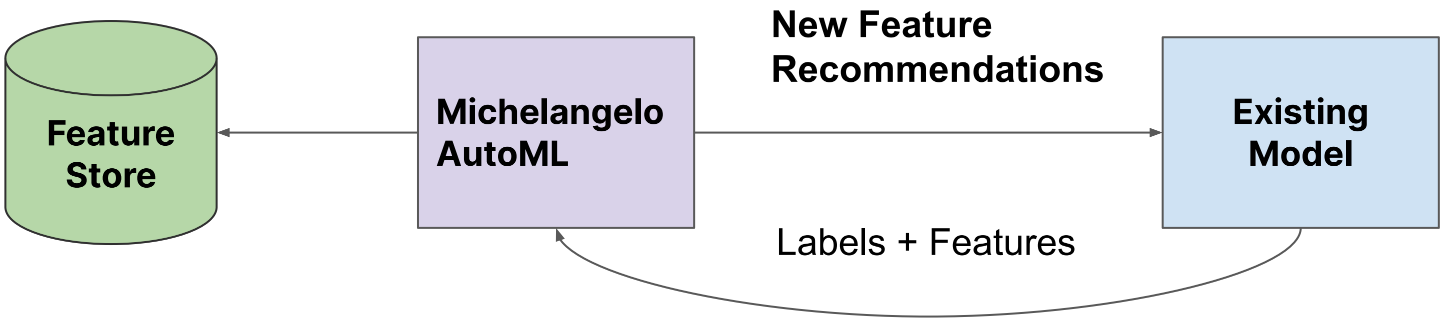
Staging tables: joins, aggregations

Feature Store tables (Point-in-time snapshots)



Feature Discovery

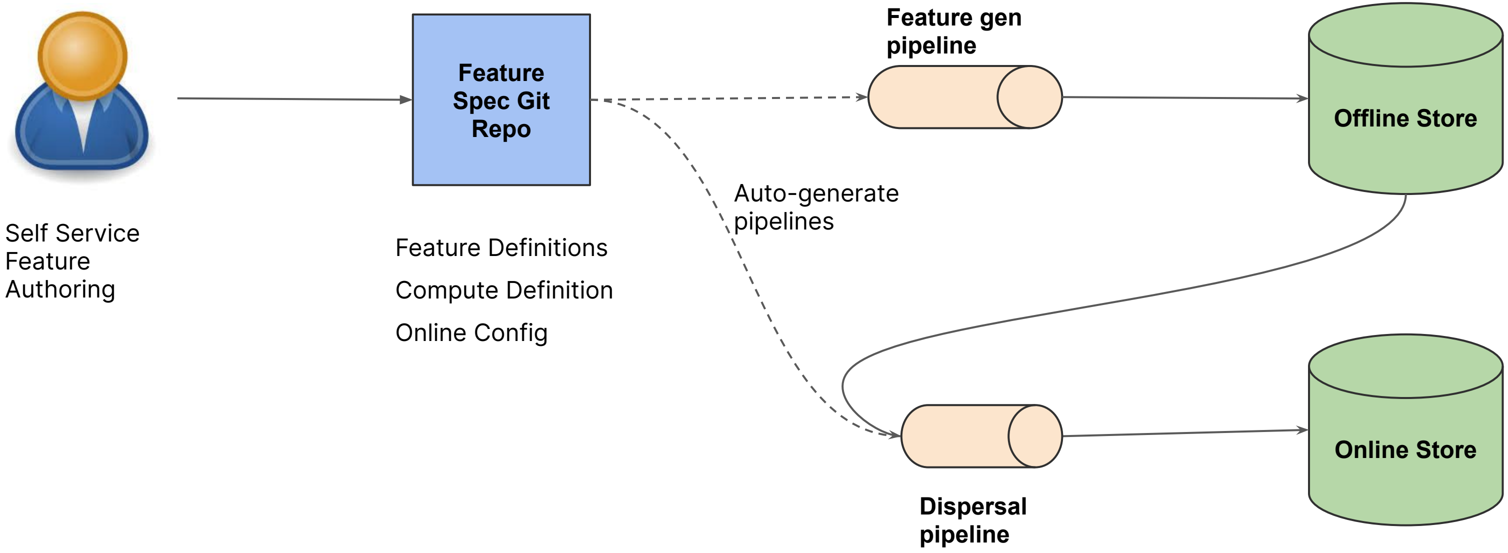
Advanced: Michelangelo AutoML tool



Boost Model Performance through Feature Discovery

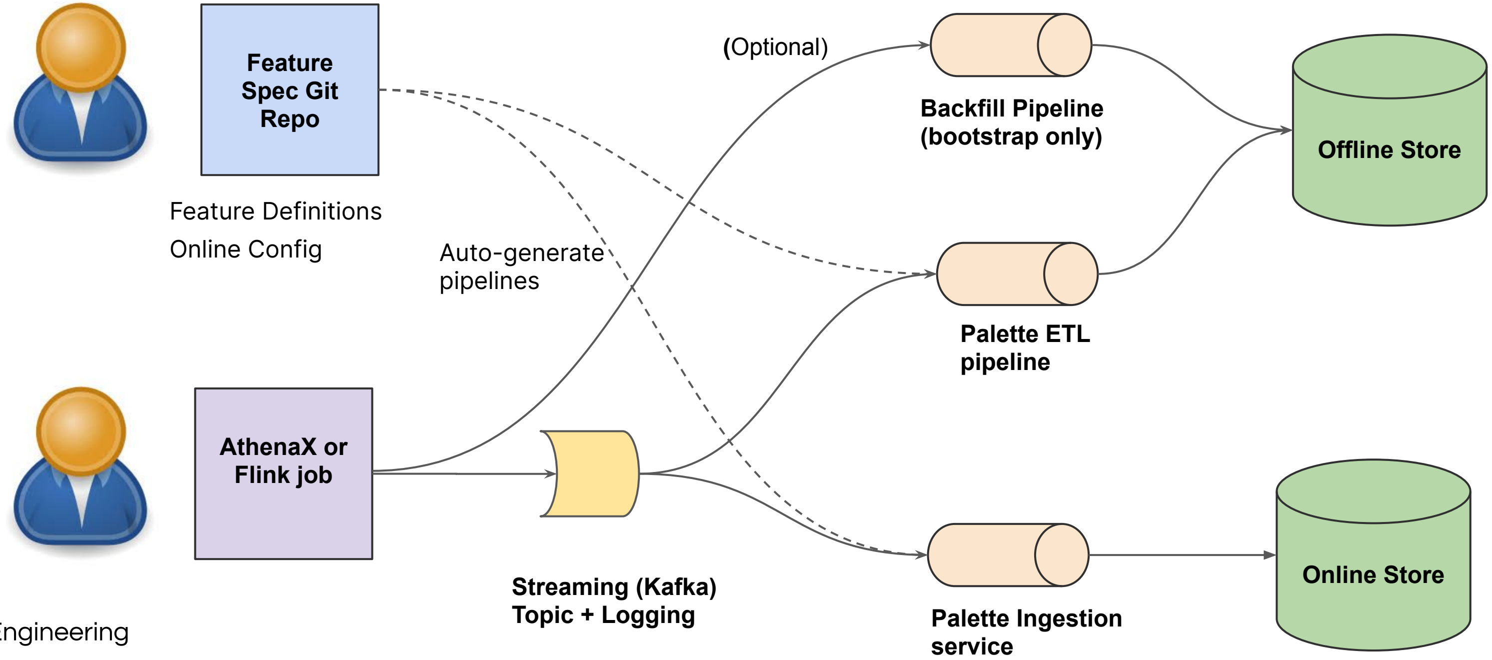
Feature Extraction

Batch computed features



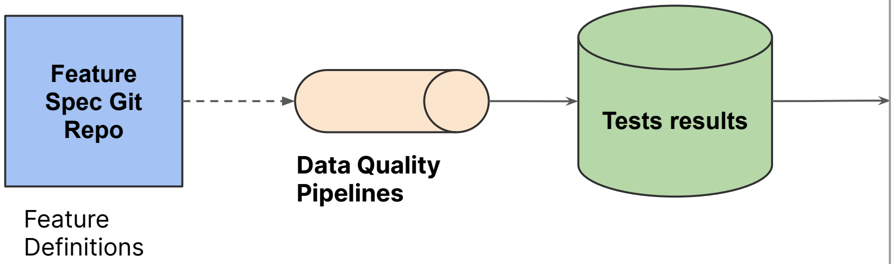
Feature Extraction

Near real-time features: automation critical for iteration velocity!



Model Training

Feature Quality: avoid debugging training time failures



Test	Date Range	Status
5 Tests	2022-06-01 - 2022-06-16	1 failure
Duplicates Pk	2022-06-01 - 2022-06-16	Failed
Cross Datacenter Consistency Rowcount		Passed
Freshness Datahawk		Passed
Missing Partitions		Passed
Null Features		Passed

At Training kickoff..

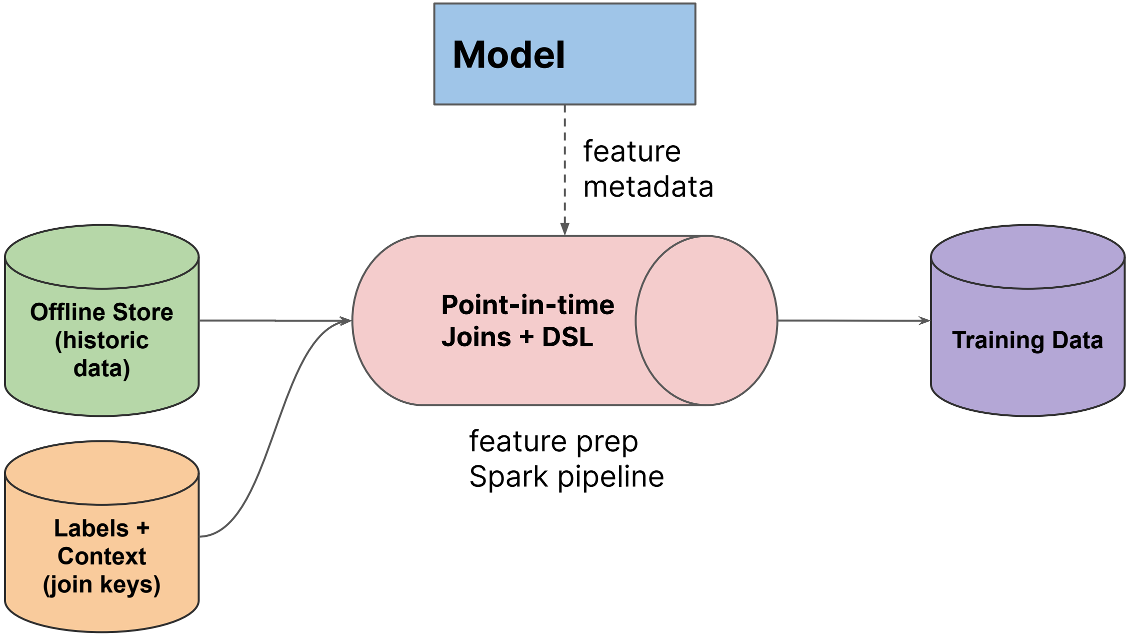
Model Training

Feature Preparation at Scale

Materialize at use: DSL for lightweight or model-specific xforms

Reducing shuffle overhead of joins is key

Incremental feature computation (Delta Store)



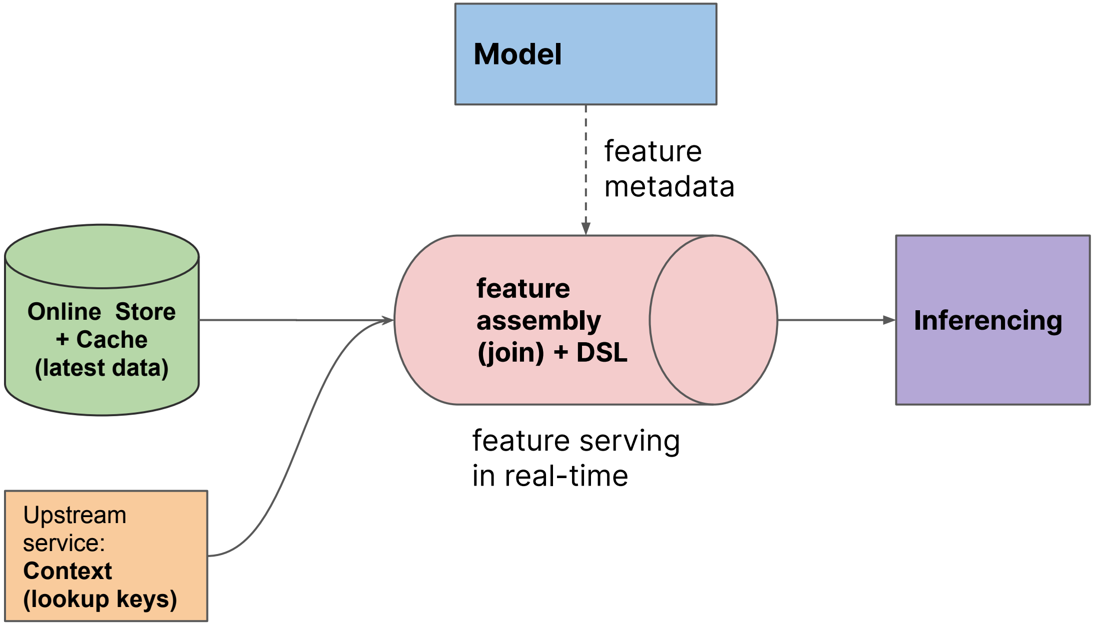
Online Model Deployment

Provisioning for Scale

Materialize at use: DSL + feature join has semantic parity to offline

Process to estimate Redis + Cassandra capacity from access patterns and feature payload

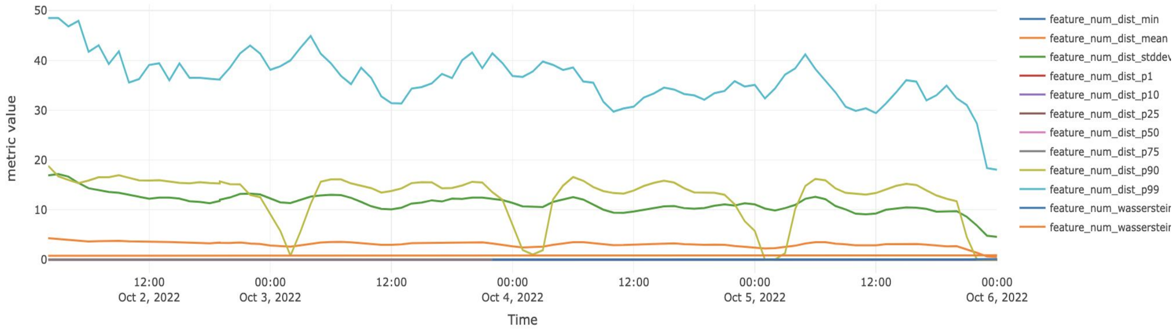
Shadow Testing through cloning Production traffic



Model Monitoring

Near real-time monitoring: use prediction logs to detect feature distribution shifts wrt to training/historic baseline

Feature Store monitoring: offline only (pre-training, pre-dispersal)



metric_name	min	max	avg	training_baseline	feature_importance
filter data...					
feature_num_dist_min	0	0	0	0	0.766606
feature_num_dist_mean	0.530115	4.285806	3.078492	25.7552	0.766606
feature_num_dist_stddev	4.556955	17.183177	11.366911	33.591095	0.766606
feature_num_dist_p1	0	0	0	0	0.766606
feature_num_dist_p10	0	0	0	0	0.766606
feature_num_dist_p25	0	0	0	19.375021	0.766606
feature_num_dist_p50	0	0	0	37.451324	0.766606

FEATURE STORE SUMMIT 2022

Feature Revision

Feature Sharing, Deprecation

Which Michelangelo models (including dev,deployed,online,offline) are using a Palette Feature?

```
# deployed_only: if set to true, only returns the online and offline models that are deployed
# palette_expression: A full or partial palette expression,
# Return projects and all the models ordered by reverse timestamp (if show_models is set to true) that are using the pa
# Note: 1. The model id prefix stands for tmYYYYMMDD-HHMMSS.
# 2. The output for the models that match the palette expression is limited to the most recent 10000

palette_expression = 'palette:eater:demo_feature'

get_models_for_features(
    deployed_only = False,
    palette_expression = palette_expression,
    show_models = False
)
```

59 Projects use this Palette Feature:

Feature → Model Lineage via metadata

Deprecate feature

No new Models
Existing Models allowed

→

Remove Feature

No Model references

Uber Engineering

Accelerating end-to-end ML

Summing it up

- Feature Discovery through search and automatic feature selection
- Feature Extraction via automated pipeline generation from spec
- Feature Preparation through data validation and scalable point-in-time joins
- Feature Serving in real-time through a process for capacity estimation, and scalable online store
- Near real-time feature monitoring via prediction logs, daily monitoring of features
- Feature sharing and deprecation via Model-Feature lineage tracking

Thank you!

Uber Engineering