Build Models Faster, and Serve Predictions at Scale, using Amazon SageMaker Feature Store

Mark Roy
Principal ML Architect
AWS

Who’s really using feature stores?

Feature store usage cuts across industries and use cases

Financial services

Consumer credit

Insurance

Online shopping

Travel

Dating

Feature Stores for ML

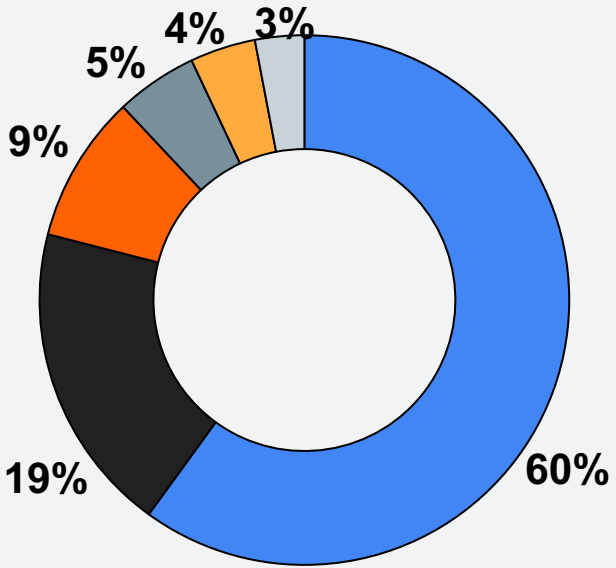
Agenda

- Feature Store overview
- Deep dive
- Demo
- Resources
- Q&A

Feature Stores for ML

Why bother using a feature store?

60% of time spent on data preparation



What data scientists spend the most time doing

- Cleaning and organizing data
- Collecting data sets
- Mining data for patterns
- Other
- Refining algorithms
- Building training sets

Source: Forbes survey of 80 data scientists, March 2016

Feature Stores for ML

… and teams too often start from scratch

Standalone feature engineering for each new model

Data sources

Vehicle damage images

Claims, customers, ...

Text interactions

Telemetry

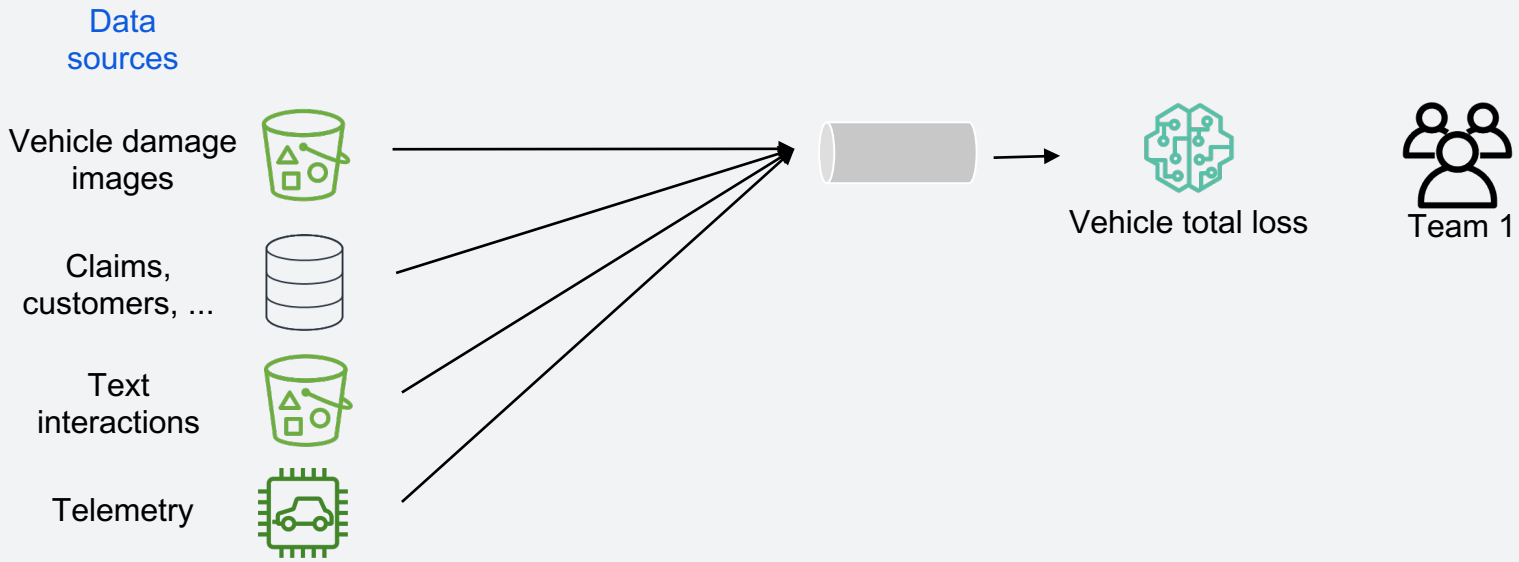
?

Vehicle total loss

Team 1

… and teams too often start from scratch

Standalone feature engineering for each new model



Feature Stores for ML

… and teams too often start from scratch

Standalone feature engineering for each new model

Data sources

Vehicle damage images

Claims, customers, ...

Text interactions

Telemetry

?

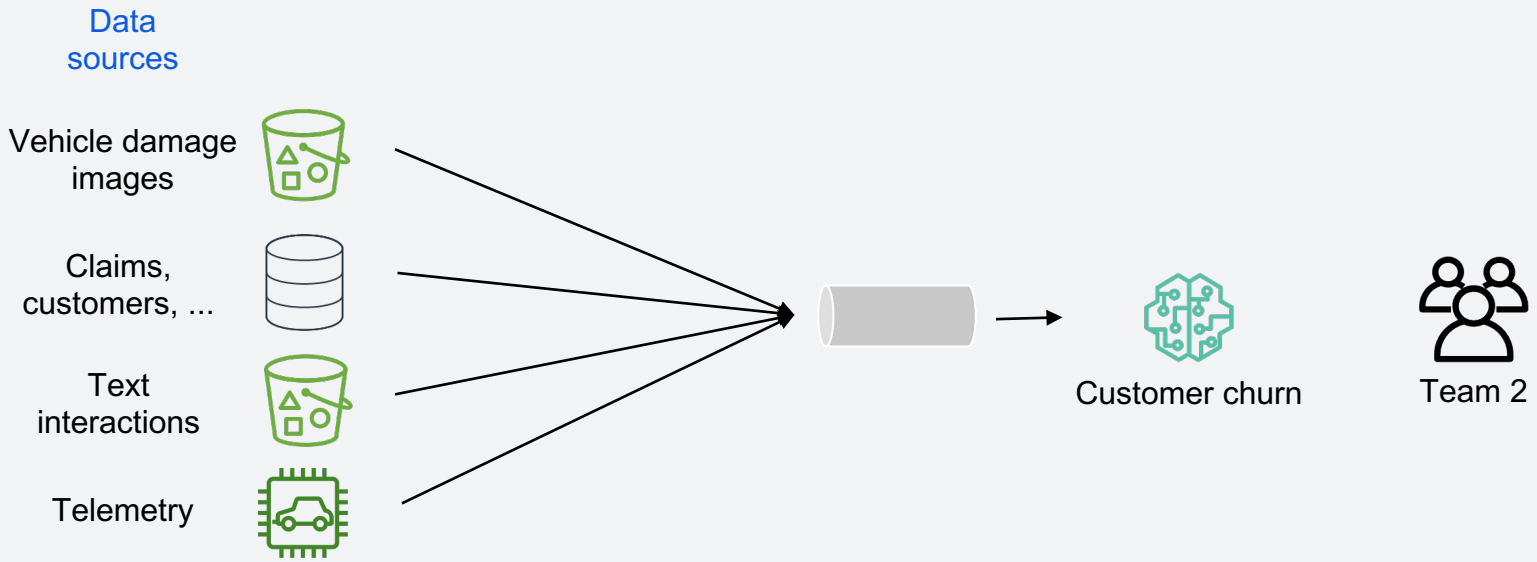
Customer churn

Team 2

Feature Stores for ML

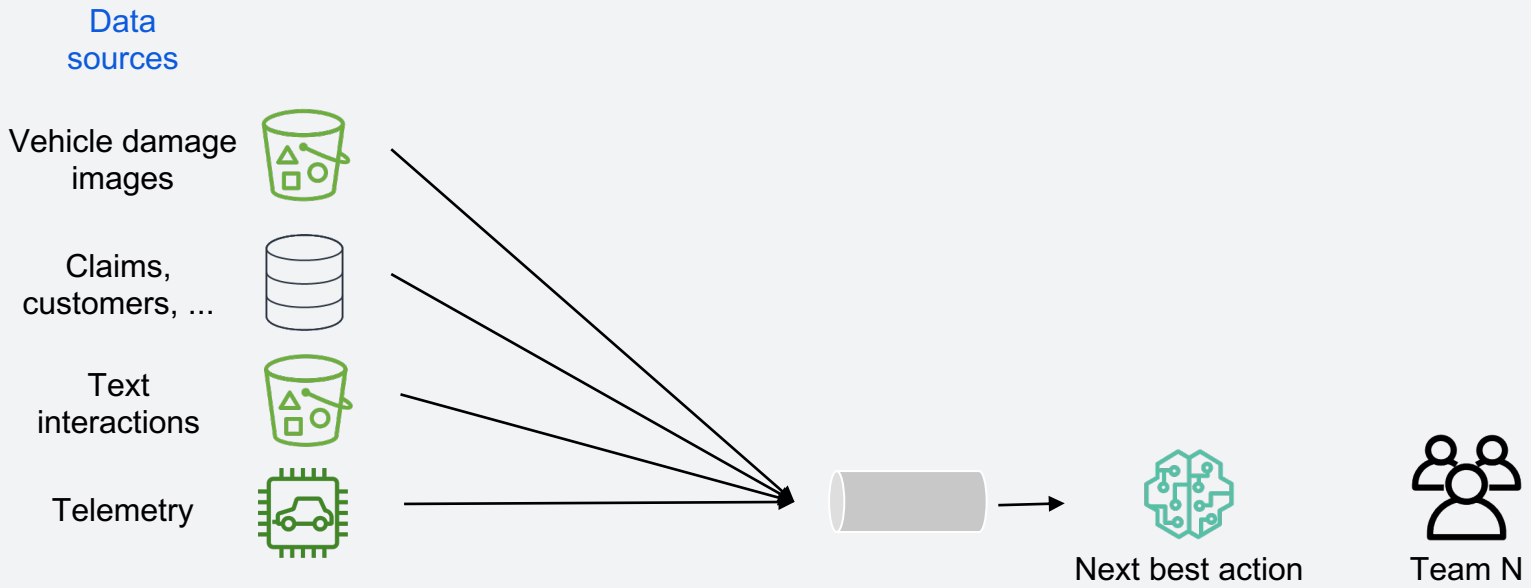
… and teams too often start from scratch

Standalone feature engineering for each new model



… and teams too often start from scratch

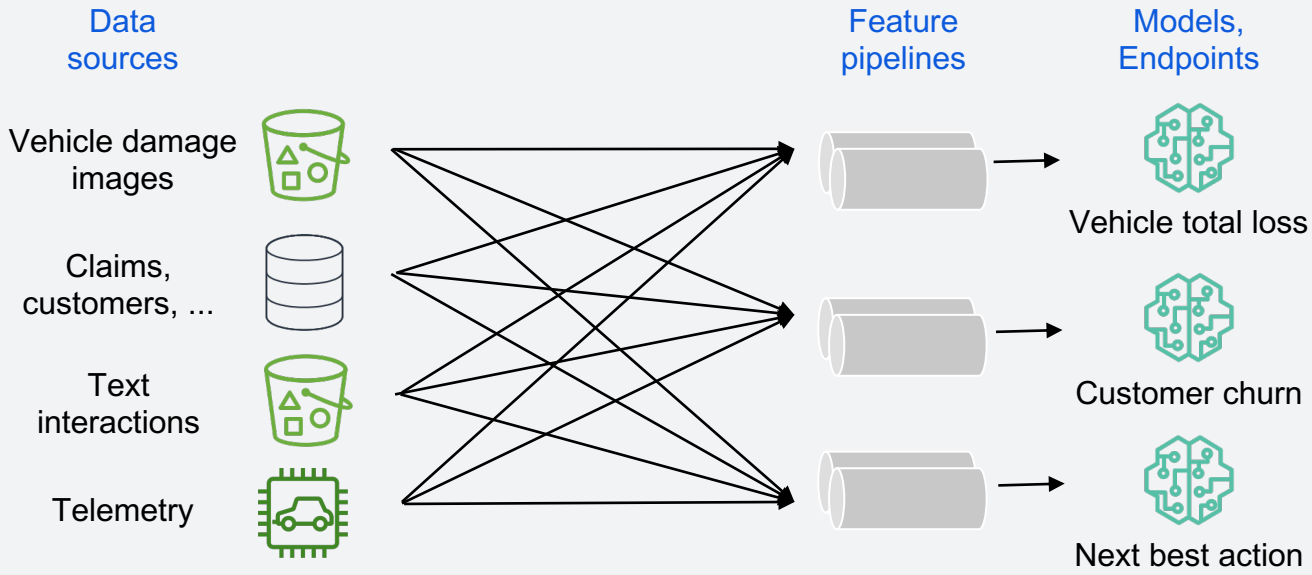
Standalone feature engineering for each new model



Feature Stores for ML

… and teams too often start from scratch

Standalone feature engineering for each new model



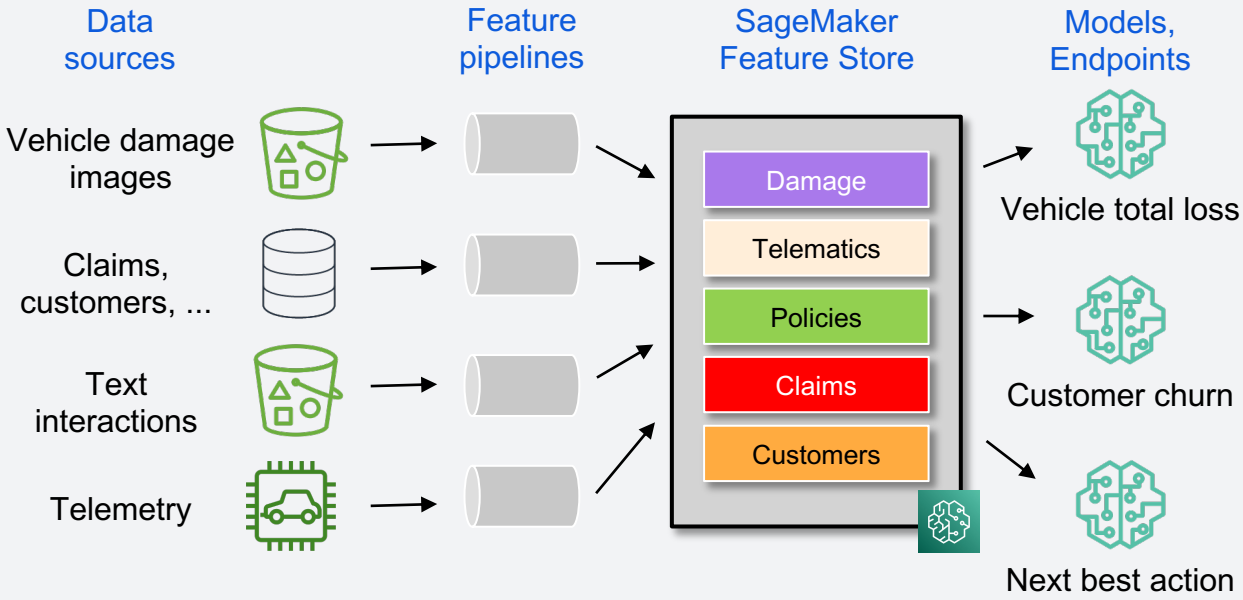
Challenges

- Slow time to market
- Feature duplication
- Inaccurate predictions

Feature Stores for ML

With SageMaker Feature Store…

Build features once, reuse them across teams and models



Benefits

- Feature reuse
- Reproducible features
- Accurate training datasets
- Low latency inference
- Consistent features for training and inference
- Managed service

Feature Stores for ML

Amazon SageMaker Feature Store

Securely store, discover, and share features for both real-time inference and training

Batch and streaming ingestion

High throughput writes for ingesting features

Online and offline features

Online features for real-time prediction, and offline features for historical data for model training and batch prediction

Feature metadata, automatic data catalog entries

Store metadata for features, and automatically create a data catalog to easily query and extract feature data

Feature discovery and reuse

Search for features to reuse, before starting new development

Security and access control

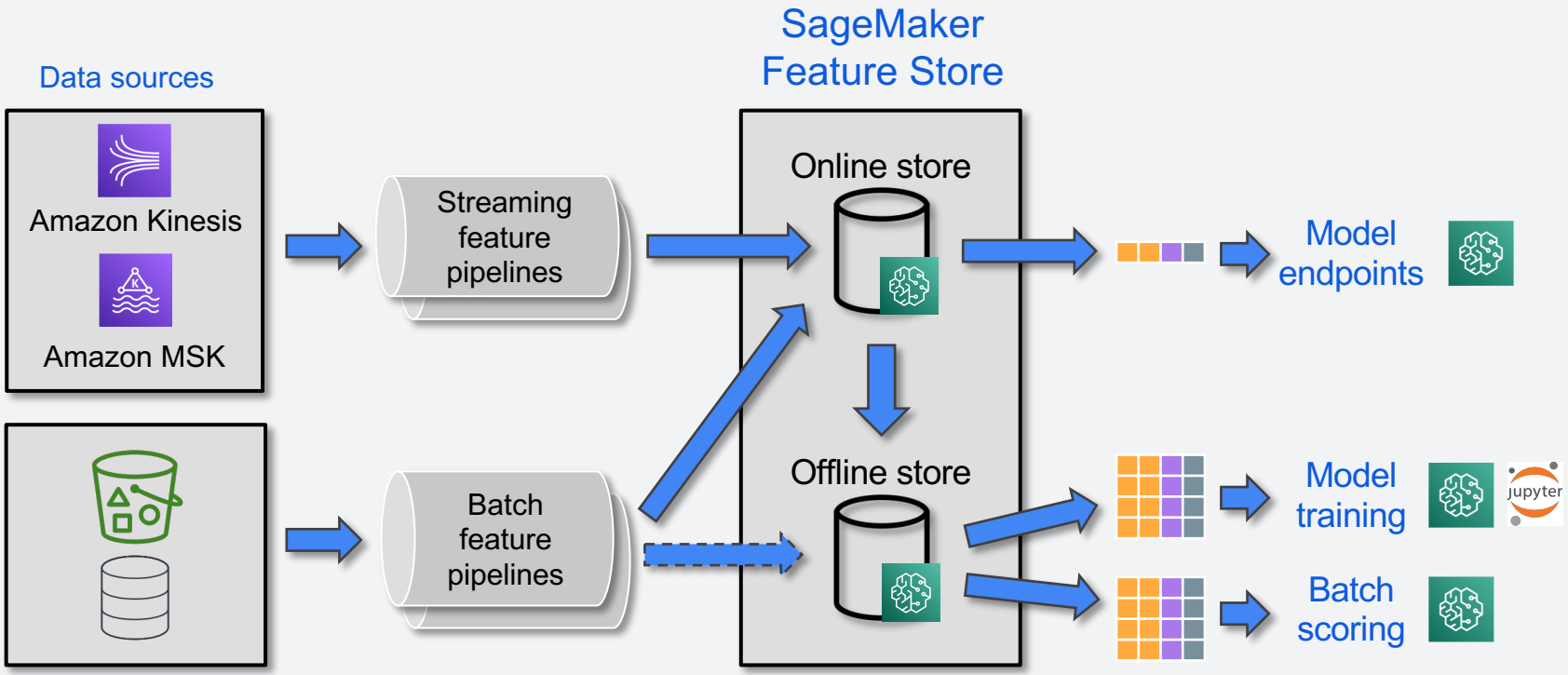
Access control for feature data and feature metadata, and support for encryption at rest, Amazon VPC, and AWS PrivateLink

Fully managed

Online features cached in low-latency store; maintain consistency between online and offline store to avoid train-infer skew

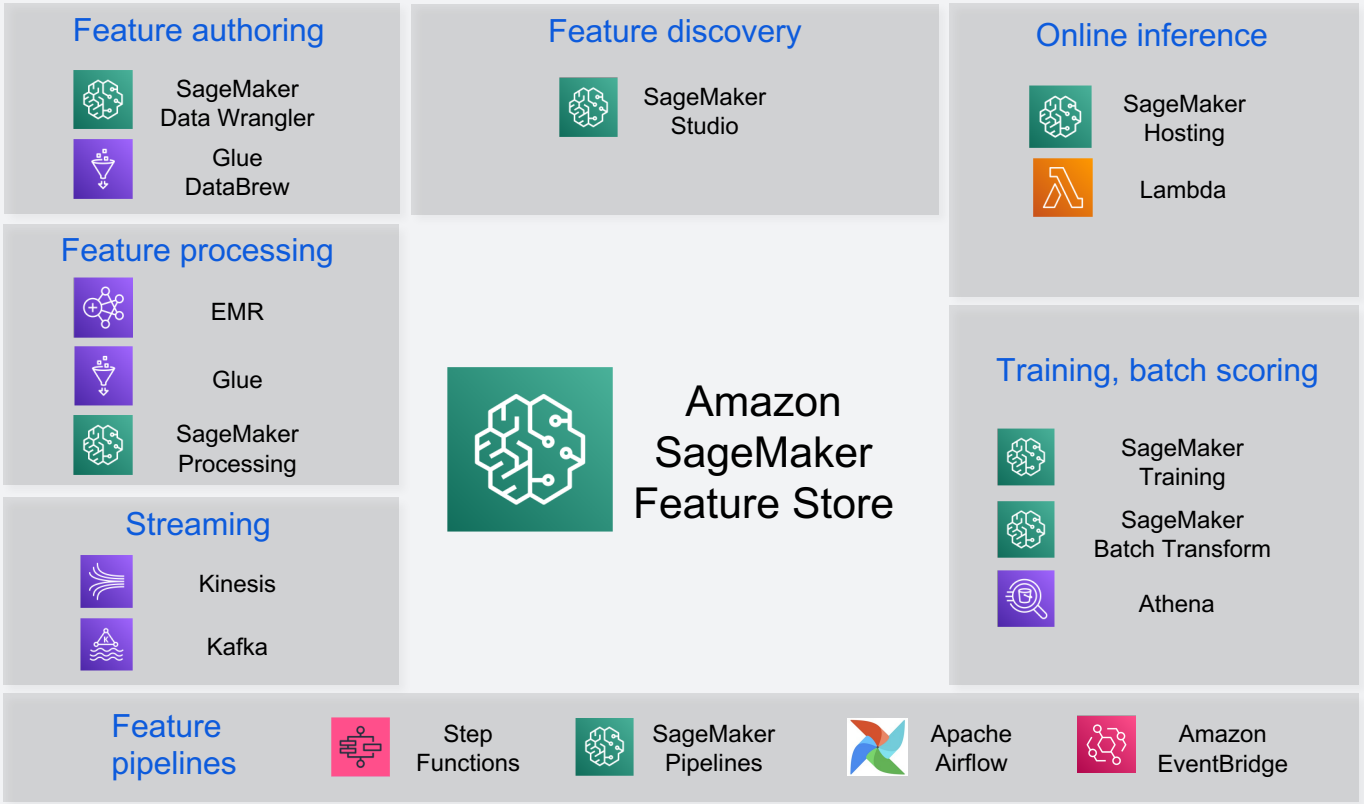
Feature Stores for ML

Feature store in context



Feature Stores for ML

Using Amazon SageMaker Feature Store with other services



Feature Stores for ML

Working with Amazon SageMaker Feature Store

- Feature ingestion
- Offline store queries
- Online feature retrieval

Feature ingestion APIs

PutRecord API

```
record = [{'FeatureName': 'feature_1', 'ValueAsString': 'val_1'},
                    …
          {'FeatureName': 'feature_N', 'ValueAsString': 'val_N'}]

sm_fs.put_record(FeatureGroupName='my-fg-name', Record=record)
```

Python SDK

```
fg = FeatureGroup(name='my-fg-name',
                  sagemaker_session=fs_session)

fg.ingest(df, max_processes=20, max_workers=4)
```

PutRecord behavior

- Online store keeps **latest feature values** for each record identifier
- Online feature values are available **immediately**, for use by any model
- Offline store appends each new record, keeping a **history of all feature values**

```
# 3 uniquely identified records
put_record('id-1', t1, 0.1)

put_record('id-2', t2, 0.2)

put_record('id-3', t3, 0.3)


# new records for existing id
put_record('id-1', t4, 0.4)

put_record('id-1', t5, 0.5)
```

Feature Group after all PutRecord calls

Online store

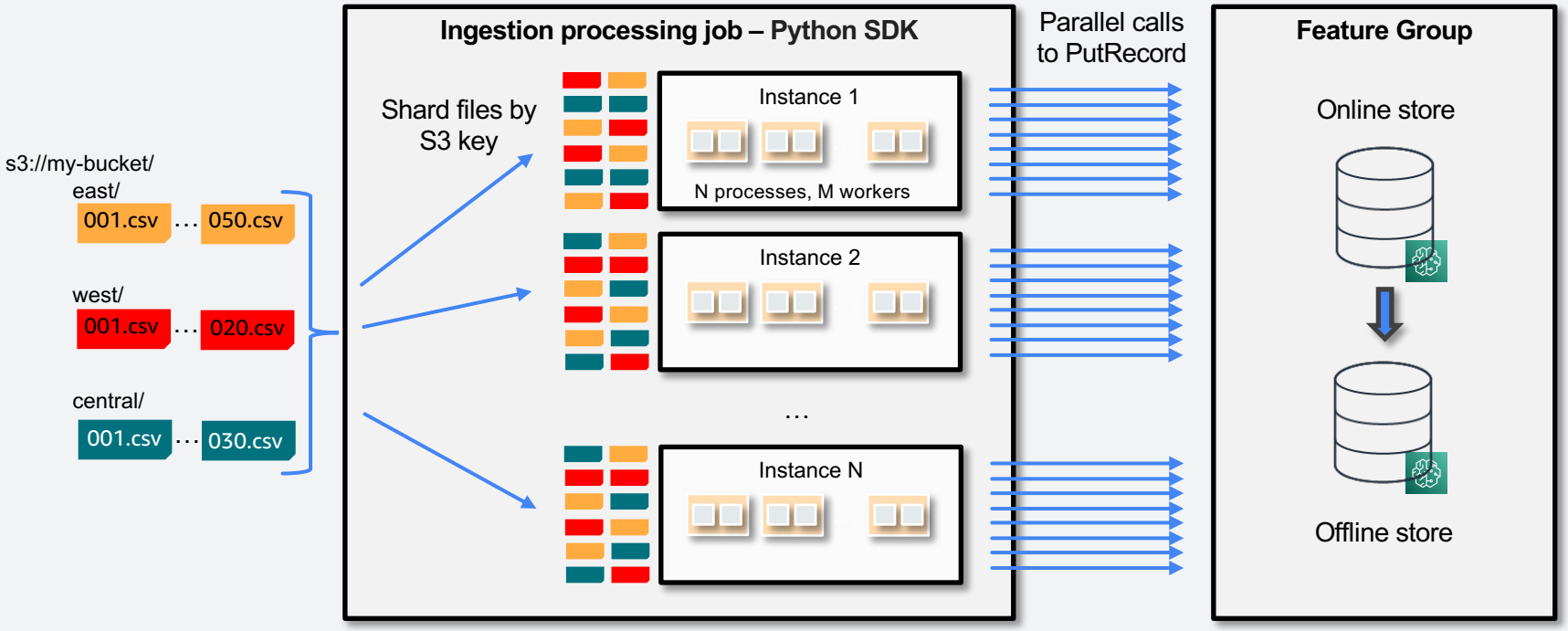
id	event time	feature
id-1	t5	0.5
id-2	t2	0.2
id-3	t3	0.3

Offline store

id	event time	feature
id-1	t1	0.1
id-2	t2	0.2
id-3	t3	0.3
id-1	t4	0.4
id-1	t5	0.5

Feature Stores for ML

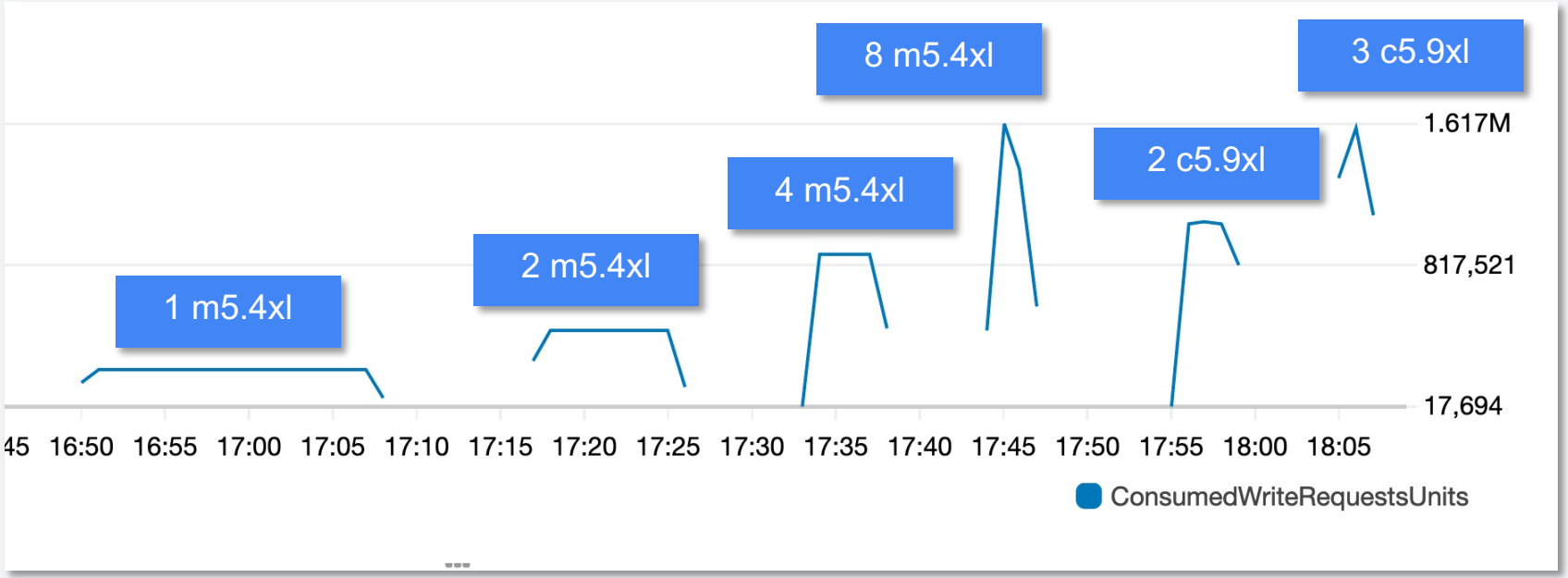
Scalable bulk ingestion with Python SDK



Feature Stores for ML

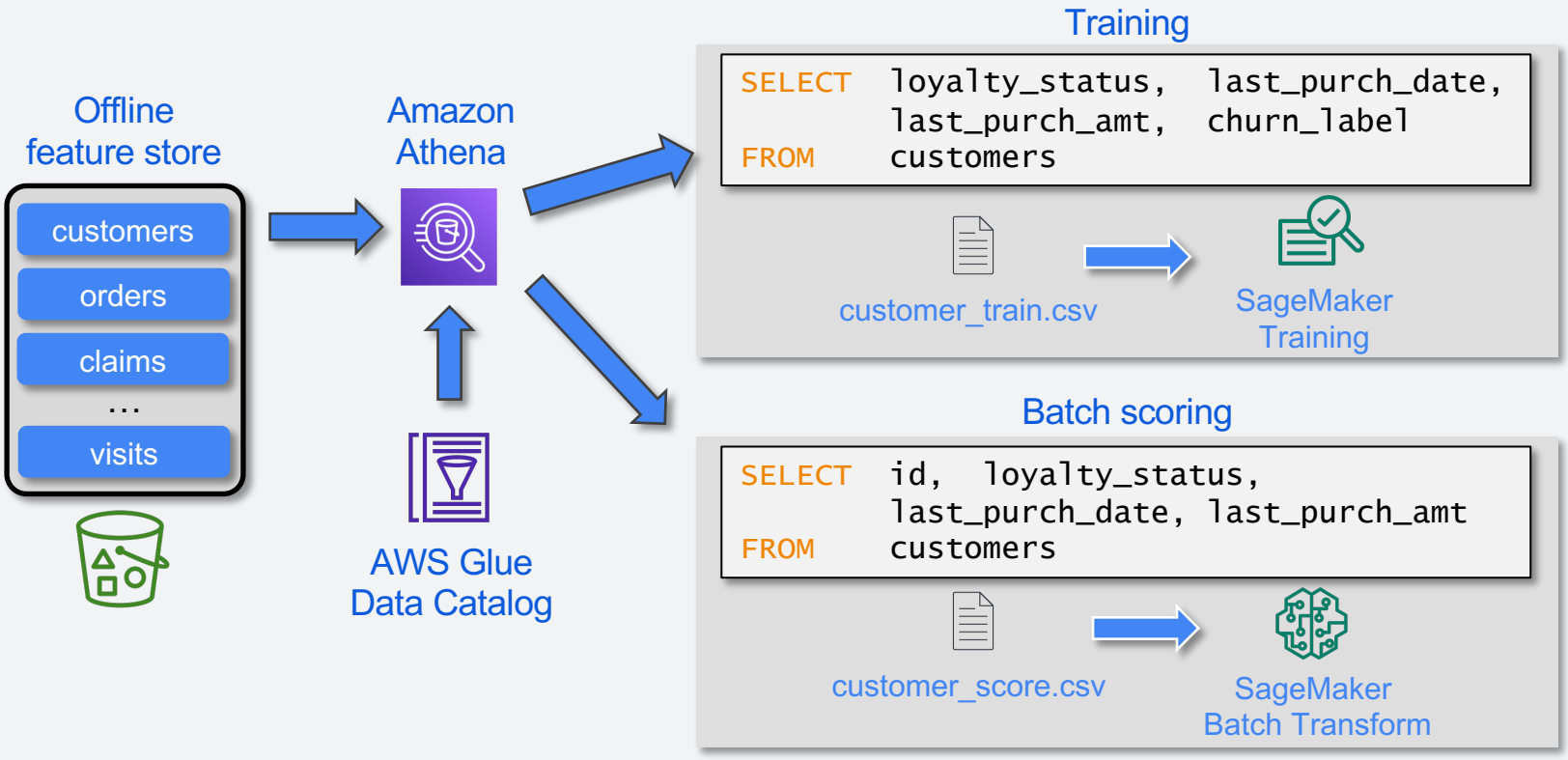
Scale up and out to speed up feature pipelines

Drive higher throughput with larger instances or more instances



Feature Stores for ML

Use SQL to create datasets from offline stores



Feature Stores for ML

Query features interactively, or with Python SDK

Athena console

```
1  select purchase_amount, is_married, p.*
2  from "customers-1628610865" c,
3       "orders-1628616113" o,
4       "products-1628610886" p
5  where c.customer_id = o.customer_id and
6        o.product_id = p.product_id
```

Ln 1, Col 1

Run | Cancel | Save as | Clear | Create

Python SDK

```
s = f'SELECT COUNT(*) FROM "{fg.athena_query().table_name}" ' + \
        'WHERE fl_date = \'2020-03-31\''
q = feature_group.athena_query()
q.run(s, output_location=output_location)
q.wait()
df = q.as_dataframe()
```

Online feature retrieval for inference

GetRecord API

```
sm_fs.get_record(FeatureGroupName='customer-fg',
                 RecordIdentifierValueAsString='CUST-001',
                 FeatureNames=['spend-last7d', 'tenure-days'])
```

BatchGetRecord API

```
sm_fs.batch_get_record(Identifiers=[
       {'FeatureGroupName': 'customer-fg',
        'RecordIdentifiersValueAsString': [
                'CUST-001', 'CUST-002', 'CUST-003'],
        'FeatureNames': ['spend-last7d', 'tenure-days']},
       {'FeatureGroupName': 'product-fg',
        'RecordIdentifiersValueAsString': [
                'P-100', 'P-200'],
        'FeatureNames': ['orders-last7d', 'daily-revenue']}
                                    ])
```

Demo

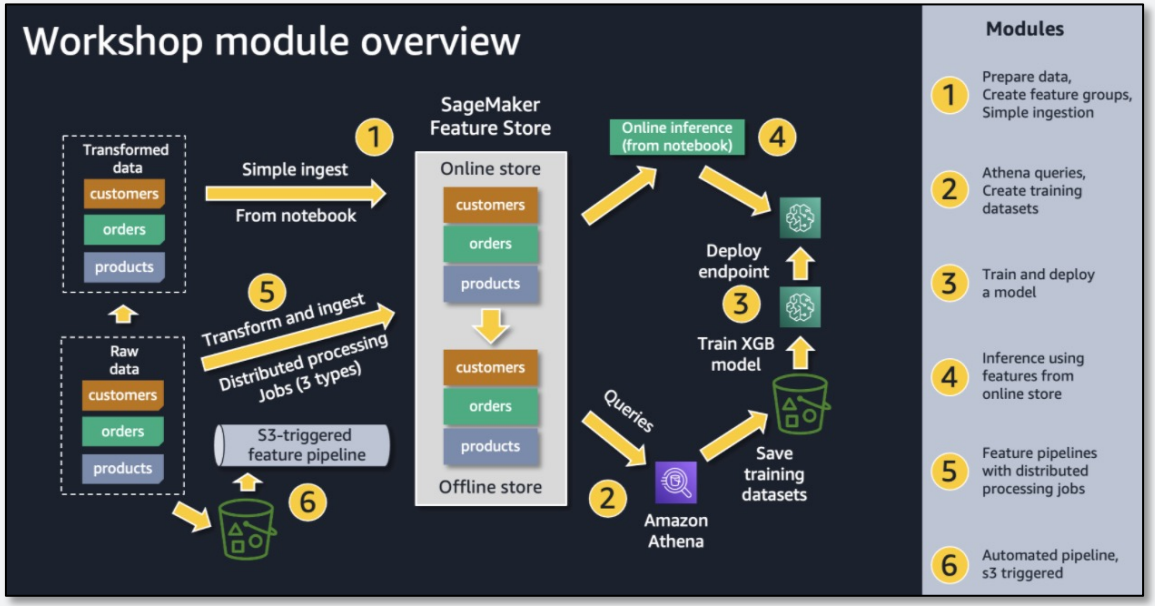
Resources

- Workshop
- Blog posts
- Documentation

Feature Stores for ML

SageMaker Feature Store workshop

New workshop repo gives you end-to-end hands-on introduction to SageMaker Feature Store - link



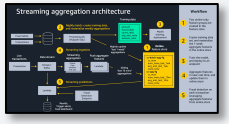
https://github.com/aws-samples/amazon-sagemaker-feature-store-end-to-end-workshop

Feature Stores for ML

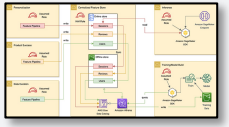
SageMaker Feature Store blog posts

Understanding key capabilities - link

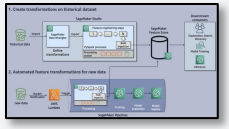
Building accurate training datasets using point-in-time queries on Apache Spark - link



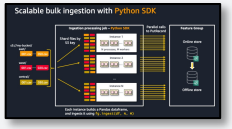
Using streaming ingestion to make ML-backed decisions in near-real time - link



Enabling feature reuse across accounts - link



Automating feature pipelines - link



Scaling batch ingestion - link

```
import random, string

filename = strftime("%Y%m%dT%H%M%SZ_", gm_time)
filename += ''.join(random.choice(string.ascii_uppercase
filename += '.parquet'

df.to_parquet(filepath + filename)
```

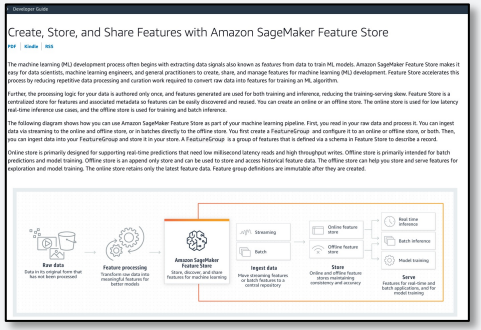
Directly ingesting historical feature data to S3 - link

```
// Create feature group
FeatureGroupOperations.createFeatureGroups(
    sageMakerClient, FEATURE_GROUP_NAMES, FEATURE_GROUP_DESCRIPTION,
    onlineStoreConfig, EVENT_TIME_FEATURE_NAME, offlineStoreConfig,
    columnDefinitions, RECORD_IDENTIFIER_FEATURE_NAME,
    SAGEMAKER_ROLE_ARN);
```

Using feature store in a Java environment - link

Feature Stores for ML

SageMaker Feature Store documentation


Developer Guide

Create, Store, and Share Features with Amazon SageMaker Feature Store

PDF | Kindle | RSS

The machine learning (ML) development process often begins with extracting data signals also known as features from data to train ML models. Amazon SageMaker Feature Store makes it easy for data scientists, machine learning engineers, and general practitioners to create, share, and manage features for machine learning (ML) development. Feature Store accelerates this process by reducing repetitive data processing and curation work required to convert raw data into features for training an ML algorithm.

Further, the processing logic for your data is authored only once, and features generated are used for both training and inference, reducing the training-serving skew. Feature Store is a centralized store for features and associated metadata so features can be easily discovered and reused. You can create an online or an offline store. The online store is used for low latency real-time inference use cases, and the offline store is used for training and batch inference.

The following diagram shows how you can use Amazon SageMaker Feature Store as part of your machine learning pipeline. First, you read in your raw data and process it. You can ingest data via streaming to the online and offline store, or in batches directly to the offline store. You first create a FeatureGroup and configure it to an online or offline store, or both. Then, you can ingest data into your FeatureGroup and store it in your store. A FeatureGroup is a group of features that is defined via a schema in Feature Store to describe a record.

Online store is primarily designed for supporting real-time predictions that need low millisecond latency reads and high throughput writes. Offline store is primarily intended for batch predictions and model training. Offline store is an append only store and can be used to store and access historical feature data. The offline store can help you store and serve features for exploration and model training. The online store retains only the latest feature data. Feature group definitions are immutable after they are created.

Raw data
Data in its original form that has not been processed

Feature processing
Transform raw data into meaningful features for better models

Amazon SageMaker Feature Store
Store, discover, and share features for machine learning

Streaming
Batch

Ingest data
Move streaming features or batch features to a central repository

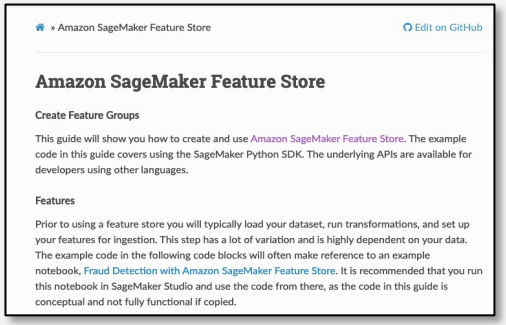
Online feature store
Offline feature store

Store
Online and offline feature stores maintaining consistency and accuracy

Real-time inference
Batch inference
Model training

Serve
Features for real-time and batch applications, and for model training

Developer guide
link


» Amazon SageMaker Feature Store

Edit on GitHub

Amazon SageMaker Feature Store

Create Feature Groups

This guide will show you how to create and use Amazon SageMaker Feature Store. The example code in this guide covers using the SageMaker Python SDK. The underlying APIs are available for developers using other languages.

Features

Prior to using a feature store you will typically load your dataset, run transformations, and set up your features for ingestion. This step has a lot of variation and is highly dependent on your data. The example code in the following code blocks will often make reference to an example notebook, Fraud Detection with Amazon SageMaker Feature Store. It is recommended that you run this notebook in SageMaker Studio and use the code from there, as the code in this guide is conceptual and not fully functional if copied.

SageMaker Python SDK
link


Boto3 Docs 1.18.19 documentation

Docs / Available services / SageMaker

TABLE OF CONTENTS

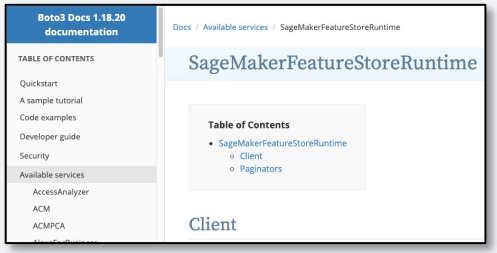
Quickstart
A sample tutorial
Code examples
Developer guide
Security
Available services
AccessAnalyzer
ACM

SageMaker

Table of Contents

- SageMaker
  - Client
  - Paginators
  - Waiters

Boto3 control plane API
link


Boto3 Docs 1.18.20 documentation

Docs / Available services / SageMakerFeatureStoreRuntime

TABLE OF CONTENTS

Quickstart
A sample tutorial
Code examples
Developer guide
Security
Available services
AccessAnalyzer
ACM
ACMPCA

SageMakerFeatureStoreRuntime

Table of Contents

- SageMakerFeatureStoreRuntime
  - Client
  - Paginators

Client

Boto3 runtime API
link

Feature Stores for ML

Thank you!

Do you have any questions?

Mark Roy

Principal ML Architect, AWS

https://www.linkedin.com/in/markproy/