



TECTON

Feature Store



Amazon SageMaker
Feature Store

featureform



Что такое Feature Store

Feature Store - это персистентное хранилище атрибутов, предназначенное для обмена атрибутами между режимами Offline, NRT, Online и Студии, характеризующееся быстрым доступом к атрибутам по ключу

Какие задачи

- Быстрый доступ к фичам в online-сценариях генерации рекомендаций
- Удобное управления фичами для исполнения моделей и движка бизнес-правил
- Переиспользование фичей между моделями
- Единые правила и способы работы с фичами в online-сценариях
- Реестр фичей и UI для работы с метаданными фичей
- Обновление фичей в NRT-режиме
- Версионирование и тегирование фичей

Какие Feature Store рассматривали

Open Source / On-premise



MLRun

featureform

SaaS / Cloud



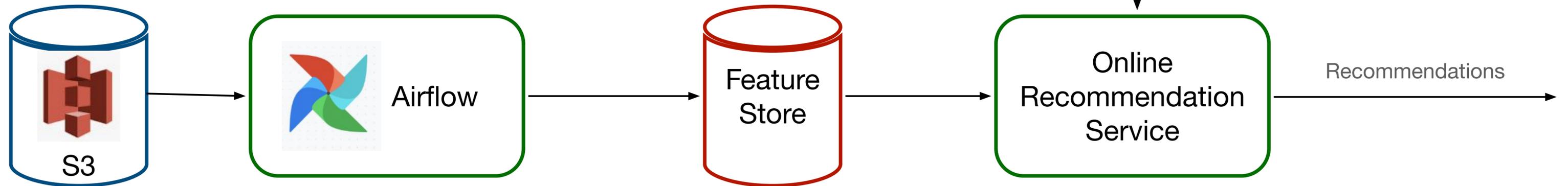
TECTON



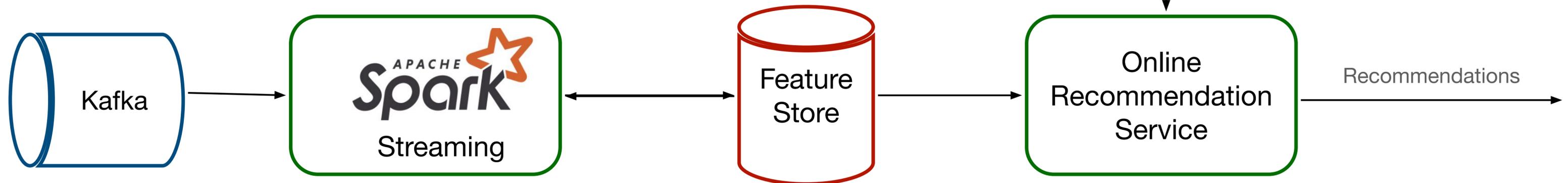
Принято решение реализовать свой
Feature Store

Сценарии использования

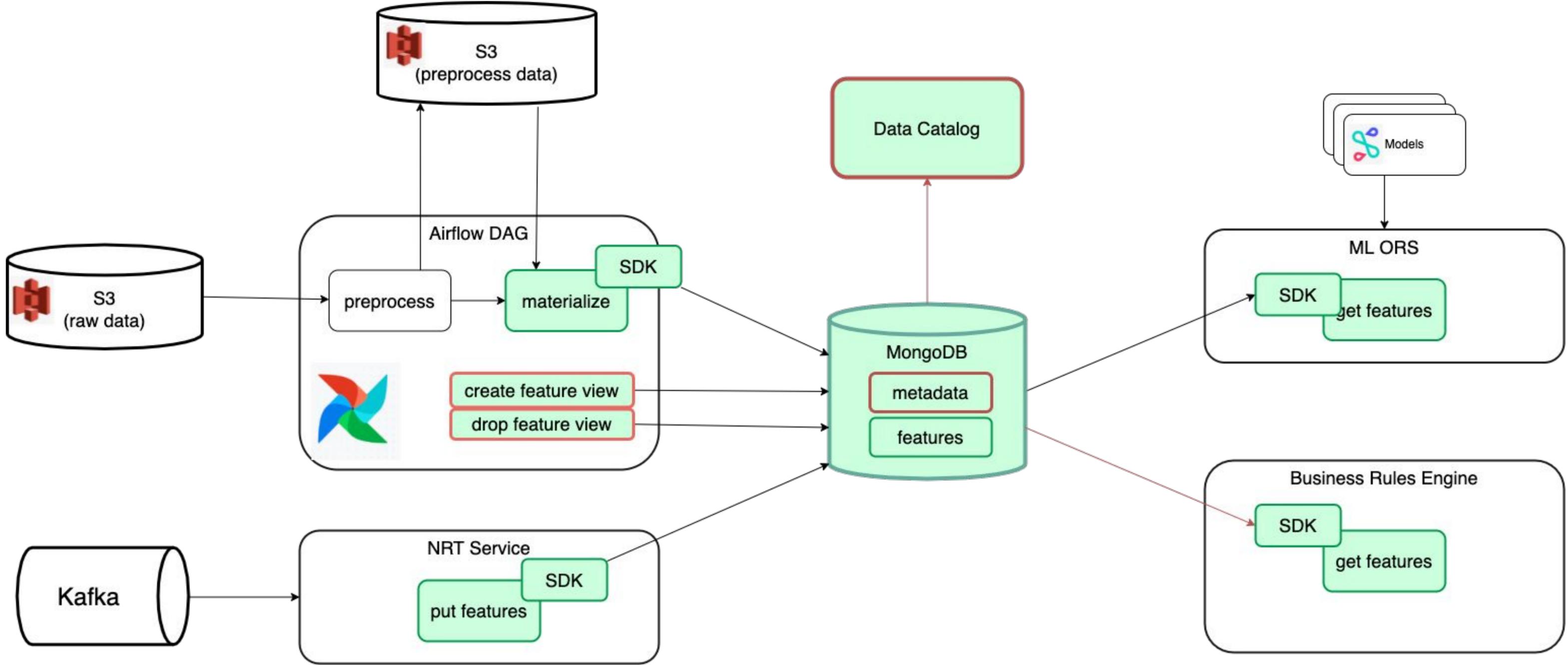
1. Периодическая генерация фичей



2. Генерация фичей в NRT-режиме



Архитектура



Feature Store на основе MongoDB + Python SDK

Концепции

Feature View

- Name
- Version
- Entity Key
- Features[]
- Metadata
- Tags[]
- Owners[]
- Description
- TTL



Feature

- Name
- Type
- Description

Python SDK:

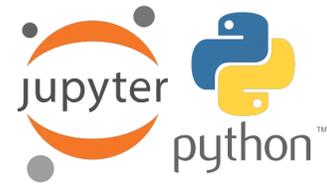
- create_feature_view()
- list_feature_view()
- describe_feature_view()
- drop_feature_view()

- materialize_features()
- put_online_features()
- get_online_features()

Customer Journey

Подготовка данных и
извлечение признаков

Разработка
фичей



Разработка пайплайна для
обновления признаков в Feature Store

Создание
Feature View



Разработка NRT
обработчика
потока данных



Разработка DAG
материализации
фичей в Feature
Store



Использование фичей
в online-сценариях

Разработка
Online
Recommendation
Service



Разработка фичей

- Experiments, Feature Engineering
- Разработка Airflow DAG по построению таблицы с фичами



| user_id | clicks_90_days | impressions_30_days | gender | country |
|----------------|-----------------------|----------------------------|---------------|----------------|
| 123 | 55 | 122 | male | RU |
| 144 | 33 | 66 | female | US |

Создание Feature View

- Включить в DAG этап создания Feature View
- Описать Feature View с помощью Python ML SDK

```
FeatureView(  
    feature_view="user_feature_view",  
    feature_view_version="v1",  
    entity_key="user_id",  
    features=[  
        Feature(name="clicks_90_days", full_name="Клики за 90 дней", description="", type=Int),  
        Feature(name="impressions_30_days", full_name="Показы за 30 дней", description="", type=Int),  
        Feature(name="gender", full_name="Пол человека", description="", type=Str),  
        Feature(name="country", full_name="Страна", description="", type=Str),  
    ],  
    ttl_days=60,  
    source={"s3_dataset": ["user_feature_view_source"], "kafka": ["users_topic"]},  
    tags=['model_1', 'best_features'],  
    producers=["dag_id"],  
    description="The best feature view you ever had about user"  
)
```

Материализация фичей

- Включить в DAG этап материализации фичей

```
df: Union[pandas.DataFrame, pyspark.sql.DataFrame] = get_data()

materialize_features(
    feature_view="user_feature_view",
    feature_view_version="v1",
    entity_key="user_id",
    dataframe=df
)
```

Обновление фичей в NRT

- Включить в NRT-обработку данных запись фичей в Feature Store
- Это можно реализовать через Spark Streaming UDF или собственный NRT-сервис

```
put_online_features(  
    entity_id=123,  
    feature_values={  
        'user_feature_streaming_view:v1:clicks_90_days': 55,  
        'user_feature_streaming_view:v1:impressions_30_days': 22,  
    },  
)
```

Получение фичей в Online Rec Service

- Включить в ORS получение фичей из Feature Store

```
feature_values = get_online_features(  
    features=[  
        "user_feature_view:v1:clicks_90_days"  
        "user_feature_view:v1:impressions_30_days"  
        "user_feature_view:v1:gender"  
        "user_feature_view:v1:country"  
    ],  
    entity_id=user_id  
)  
  
return model_predict(  
    model_name='best_model',  
    model_version='1',  
    feature_values=feature_values)
```

План

На релиз:

- Хранилище на основе MongoDB
- Python SDK:
 - `materialize_features()`
 - `get_features()`
 - `put_features()`

После релиза:

- Работа с метаданными FeatureView
- Валидация метаданных
- Интеграция с Data Catalog
- Интеграция с движком бизнес-правил
- UI по просмотру метаданных
- TTL данных

Общее сравнение

| | Feast | Hopsworks | MLRun | FeatureForm |
|----------------------------------|--|---|--|---|
| Feature Catalog / Web UI | + | + | + | + |
| Data Sources (S3, Kafka) | <ul style="list-style-type: none"> • S3 • PostgreSQL • Spark (alpha) • Kafka | <ul style="list-style-type: none"> • S3 • HopsFS | <ul style="list-style-type: none"> • S3 • PostgreSQL | <ul style="list-style-type: none"> • S3 • PostgreSQL |
| Offline Feature Store | <ul style="list-style-type: none"> • S3 • Spark (alpha) | <ul style="list-style-type: none"> • S3 • HopsFS | <ul style="list-style-type: none"> • S3 • PostgreSQL | <ul style="list-style-type: none"> • S3 • PostgreSQL |
| Online Feature Store | <ul style="list-style-type: none"> • Redis • PostgreSQL • Cassandra | <ul style="list-style-type: none"> • RonDB (MySQL Cluster) | <ul style="list-style-type: none"> • in-memory dicts | <ul style="list-style-type: none"> • Redis • PostgreSQL • Cassandra • EmbeddingsHub |
| KNN/ANN in Online FS | - | - | - | <ul style="list-style-type: none"> • EmbeddingsHub (RocksDB + hnswlib) (single-node) |
| Feature Lineage / Transformation | - | + | + | + |
| Feature tags / versioning | + | + | + | + |
| Model Feature Tracking | + | + | + | + |
| Customizing ability | + | - | - | - |
| Security | - | +/- | - | - |

Почему не подошел Feast

1. Нет интеграции с MongoDB
2. Невозможно делать поиск по значению фичей (нужно для движка бизнес правил)
3. Интеграция с Kafka поддерживается только через Spark Streaming
4. Фичи работы в NRT режиме и работы с Spark находятся в Alpha версии и не поддерживаются Maintainer
5. Сравнение по скорости Feast + Redis с собственным решением на Redis показало, что собственная реализация в 3 раза быстрее
6. Чтобы выгрузить датасет с фичами - необходимо подать на вход все Entity ID, что довольно неудобно
7. Невозможно использовать для event-подобных данных (для оффлайн стора), только для классических фичей-характеристик сущностей

Примеры фичей

| rails_user_features→ | make_train_dataset | rails_item_features→ | make_train_dataset | rails_pair_features→ | make_train_dataset |
|----------------------------|----------------------------|----------------------------|----------------------------|---------------------------|---------------------------------------|
| user_uid | age_access_type_0 | collection_uid | age_access_type_0 | user_uid | frame_30_sum_n_pagings |
| age_access_type_0 | age_access_type_12 | age_access_type_0 | age_access_type_12 | rail_uid | frame_30_sum_n_selects |
| age_access_type_12 | age_access_type_16 | age_access_type_12 | age_access_type_16 | date | frame_30_sum_n_watches |
| age_access_type_16 | age_access_type_18 | age_access_type_16 | age_access_type_18 | week_date | frame_30_sum_watched_time |
| age_access_type_18 | age_access_type_6 | age_access_type_18 | age_access_type_6 | watched_time | frame_30_max_pos |
| age_access_type_6 | country_france | age_access_type_6 | country_france | is_active | frame_30_cr_stow |
| country_france | country_germany | country_france | country_germany | frame_30_sum_n_pagings | frame_30_awtpp |
| country_germany | country_other | country_germany | country_other | frame_30_sum_n_selects | frame_90_sum_n_pagings |
| country_great-britain | country_russia | country_great-britain | country_russia | frame_30_sum_n_watches | frame_90_sum_n_selects |
| country_other | genre_287a1485-7a88-4c2f-b | country_other | genre_287a1485-7a88-4c2f-b | frame_30_sum_watched_time | frame_90_sum_n_watches |
| country_russia | genre_Action | country_russia | genre_Action | frame_30_max_pos | frame_90_sum_watched_time |
| country_usa | genre_Adventure | country_usa | genre_Adventure | frame_90_sum_n_pagings | frame_90_max_pos |
| genre_287a1485-7a88-4c2f-b | genre_Comedy | genre_287a1485-7a88-4c2f-b | genre_Comedy | frame_90_sum_n_selects | frame_90_cr_stow |
| genre_Action | genre_Drama | genre_Action | genre_Drama | frame_90_sum_n_watches | frame_90_awtpp |
| genre_Adventure | genre_Family | genre_Adventure | genre_Family | frame_90_sum_watched_time | |
| genre_Comedy | genre_other | genre_Comedy | genre_other | frame_90_max_pos | |
| genre_Drama | genre_Sci-Fi | genre_Drama | genre_Sci-Fi | | |
| genre_Family | genre_Thriller | genre_Family | genre_Thriller | | |
| genre_Sci-Fi | release_year_1980 | genre_Sci-Fi | mean_rating | | |
| genre_Thriller | release_year_1990 | genre_Thriller | n_titles | n_pagings | Кол-во пролистываний карточек фильмов |
| genre_other | release_year_2000 | genre_other | release_year_1980 | n_selects | Кол-во заходов на карточку |
| release_year_1980 | release_year_2010 | release_year_1980 | release_year_1990 | n_watches | Кол-во просмотров |
| release_year_1990 | release_year_2015 | release_year_1990 | release_year_2000 | watched_time | Время просмотра |
| release_year_2000 | release_year_2017 | release_year_2000 | release_year_2010 | max_pos | Максимальный заход по пролистываниям |
| release_year_2010 | release_year_2020 | release_year_2010 | release_year_2015 | cr_stow | $n_watches/n_selects$ |
| release_year_2015 | svod_ratio | release_year_2015 | release_year_2017 | awtpp | $watched_time/pagings$ |
| release_year_2017 | user_watch_count | release_year_2017 | release_year_2020 | svod_ratio | Доля просмотров по подписке |
| release_year_2020 | | release_year_2020 | svod_ratio | | |
| svod_ratio | | svod_ratio | | | |
| user_watch_count | | mean_rating | | | |
| | | n_titles | | | |