AWS re:Invent

AIM358

Prepare data for ML using Amazon SageMaker

Christian Williams
Machine Learning Specialist SA
Amazon Web Services

AWS re:Invent

aws

Agenda

- What problem are we solving?
- What are Amazon SageMaker inference pipelines?
- Use cases for Amazon SageMaker inference pipelines
- Q & A

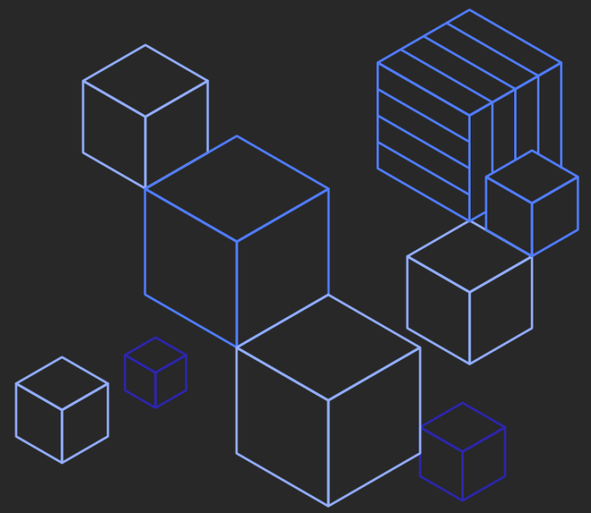
Related breakouts

AIM348 - Deploying and managing machine learning models at scale

AIM306 - How to build high-performance machine learning solutions at low cost

AIM416 - Deploy an ML model on the cloud and at the edge

AIM357 - Build an ETL pipeline to analyze customer data



What problem are we solving?

Just like cheese and vegetables

Data comes in many forms

- **Data may need pre-processing**

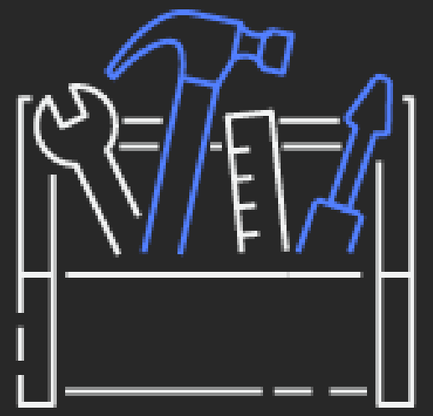
  Normalization, feature engineering, dimensionality reduction, etc.

- **Predictions may need post-processing**

  Filtering, sorting, combining, etc.

- **Prior to inference pipelines**

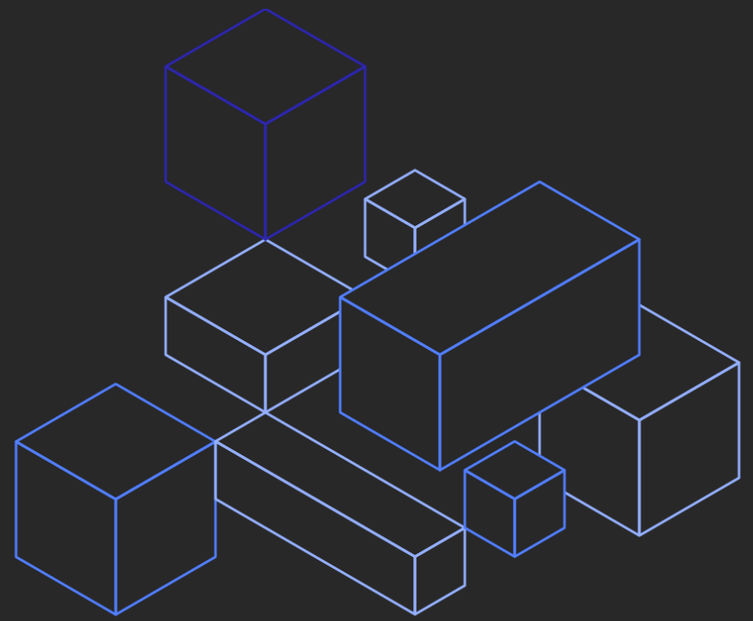
  This required deploying multiple endpoints



Tell me: What is this pipeline?

Tell me: What is this pipeline?

- Linear sequence of 2-5 containers that process inference requests
- Feature engineering with Scikit-learn or SparkML (on AWS Glue or Amazon EMR)
- Predict with built-in or custom containers
- The pipeline is deployed as a single model



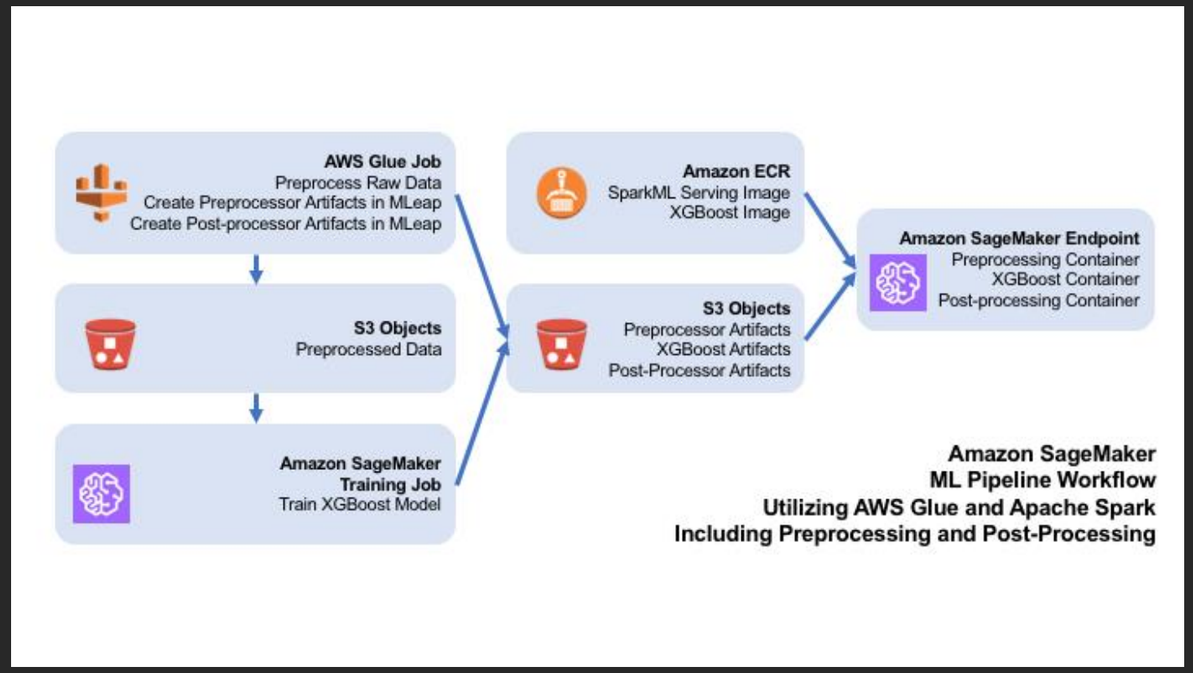
SparkML/XGBoost use case

- Using AWS Glue for executing the SparkML feature pre-processing and post-processing job

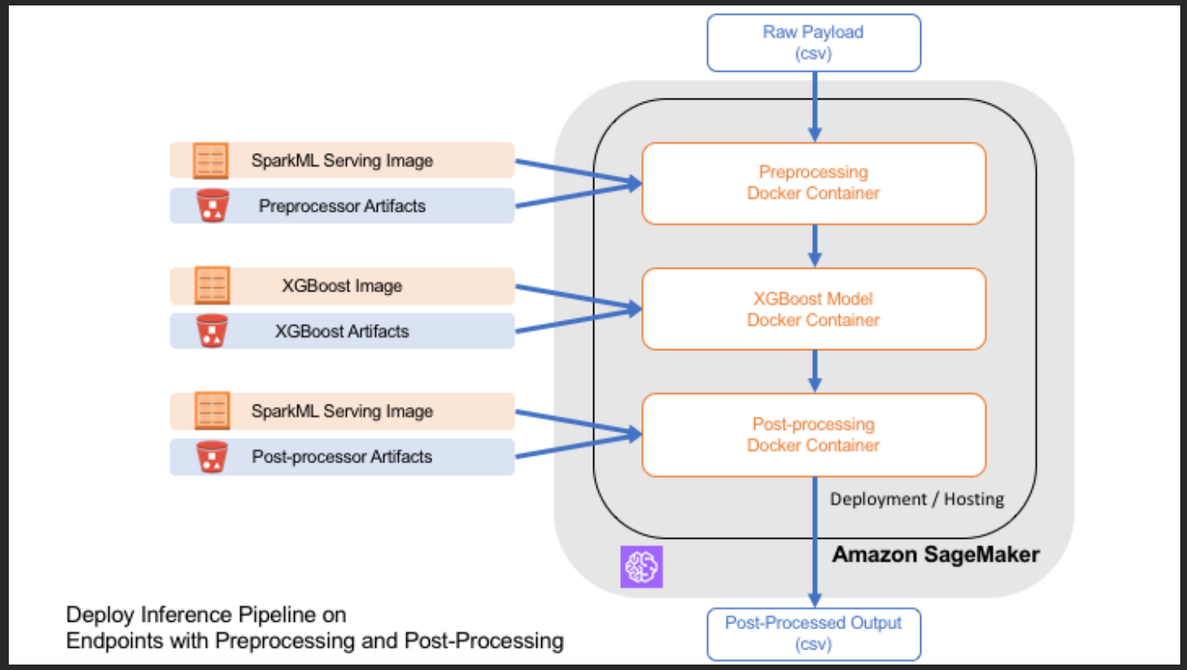
- Using Amazon SageMaker XGBoost to train on the processed dataset produced by SparkML job

- Building an inference pipeline consisting of SparkML & XGBoost models for a real-time inference endpoint

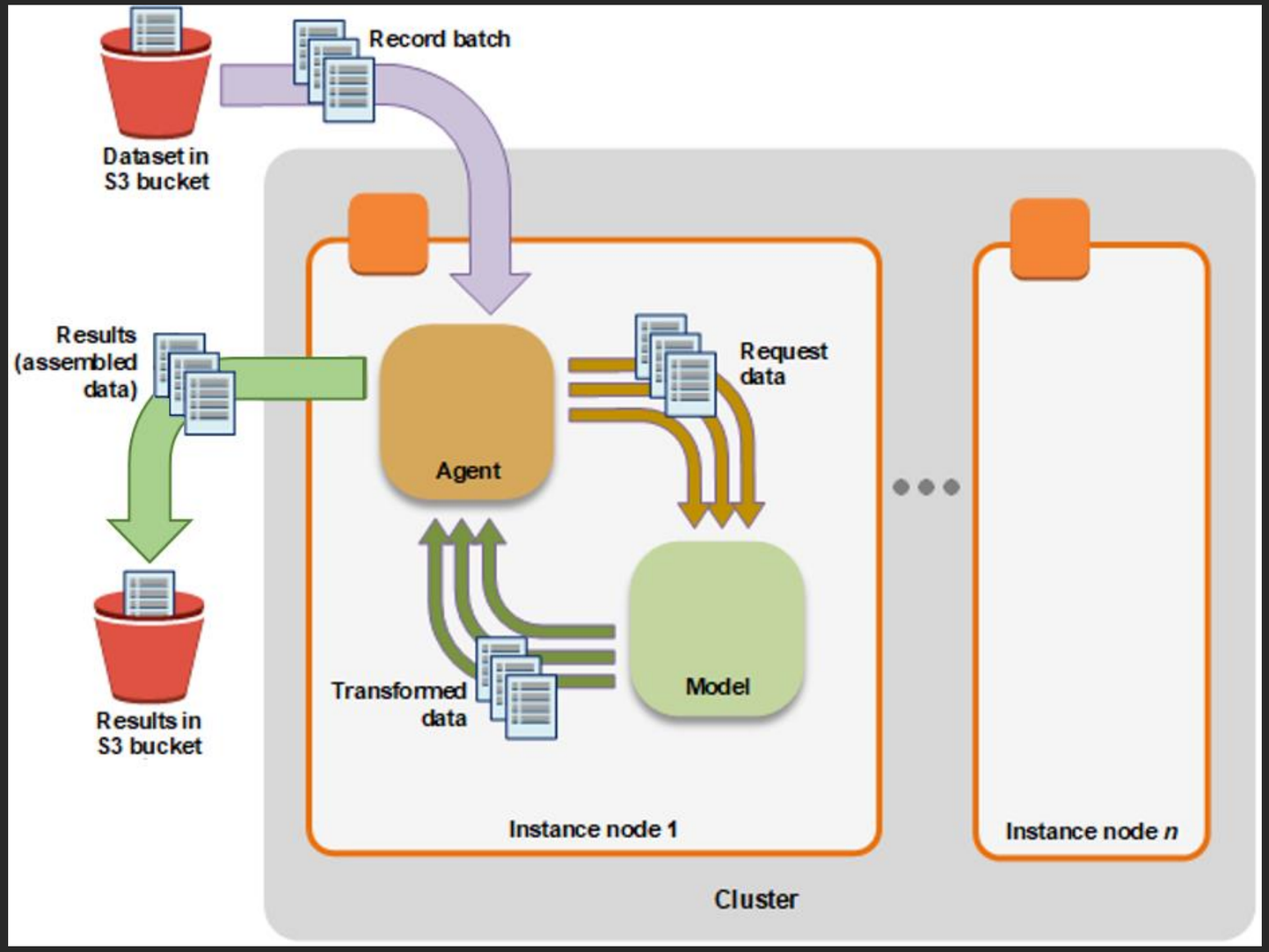
Workflow



Endpoint



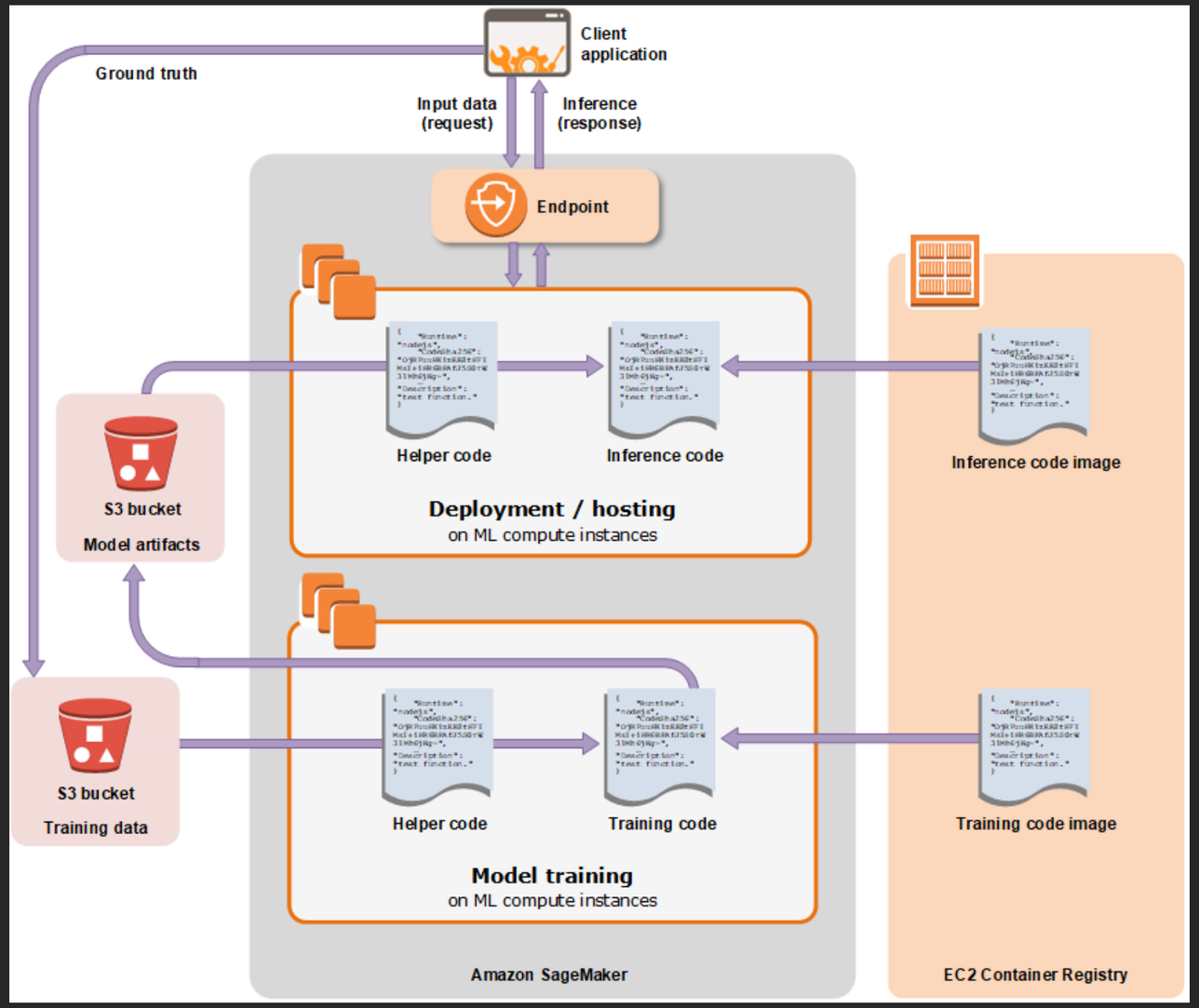
Batch transform



Batch transform

```
input_data_path = 's3://{}/{}/{}'.format(default_bucket, 'key', 'file_name')
output_data_path = 's3://{}/{}'.format(default_bucket, 'key')
transform_job = sagemaker.transformer.Transformer(
    model_name = model_name,
    instance_count = 1,
    instance_type = 'ml.m4.xlarge',
    strategy = 'SingleRecord',
    assemble_with = 'Line',
    output_path = output_data_path,
    base_transform_job_name='inference-pipelines-batch',
    sagemaker_session=sess,
    accept = CONTENT_TYPE_CSV)
transform_job.transform(data = input_data_path,
                        content_type = CONTENT_TYPE_CSV,
                        split_type = 'Line')
```

Real-time inference



Real-time inference

```
from sagemaker.predictor import json_serializer, json_deserializer, RealTimePredictor
from sagemaker.content_types import CONTENT_TYPE_CSV, CONTENT_TYPE_JSON

payload = {
        "input": [
            {
                "name": "Pclass",
                "type": "float",
                "val": "1.0"
            },
            ...
        ],
        "output": {
            "name": "features",
            "type": "double",
            "struct": "vector"
        }
    }

predictor = RealTimePredictor(endpoint=endpoint_name, sagemaker_session=sess, serializer=json_serializer,
                              content_type=CONTENT_TYPE_JSON, accept=CONTENT_TYPE_CSV)

print(predictor.predict(payload))
```

That’s easy enough: Let’s look at an example

Demo

Thank you!

AWS re:Invent

aws

Please complete the session survey in the mobile app.