yieldmo

Case study

A scalable, configuration-driven Machine Learning Platform

www.griddynamics.com

Grid Dynamics

Contents

Introduction

Yieldmo, a Grid Dynamics client, is an advertising platform that helps brands improve digital ad experiences through creative tech and artificial intelligence (AI). The company uses bespoke ad formats, proprietary attention signals, predictive format selection, and privacy-safe inventory curation. Yieldmo believes all ads should captivate users and be tailored to their liking. It helps brands deliver the best ad for every impression opportunity. Thanks to its advances in AI, its proprietary measurement technology, and its close relationships with publishers, this vision is increasingly attainable.

COMPANY

Yieldmo

INDUSTRY

Advertising

REGION

United States

The challenge

Yieldmo operates on massive amounts of data and processes billions of ad placements daily. Connecting publishers and advertisers requires a highly scalable platform with low-latency response, as well as the ability to match users, publishers, and advertisers to optimize campaign delivery and performance. To curate the ad inventory for the various needs of our customers, Yieldmo's ad server has to assess in real time how likely a particular ad request is to result in a desired event, such as a click or video completion, for a particular ad campaign or an advertiser. Having such capabilities requires sophisticated machine learning (ML) models and advanced software engineering to develop and deliver ML models to production.

ML model development is an iterative process, proceeding from creating datasets to feature engineering, training models, and creating model artifacts that are then used for making predictions, which, in turn, guide decisions for ad serving in real time. Scaling a machine learning platform calls for proper data engineering and data management along with a clear path from the code and model settings developed by data scientists in a research environment to automated data pipelines and prediction deployment in production.

System requirements

At a very high level, the following requirements should be met:

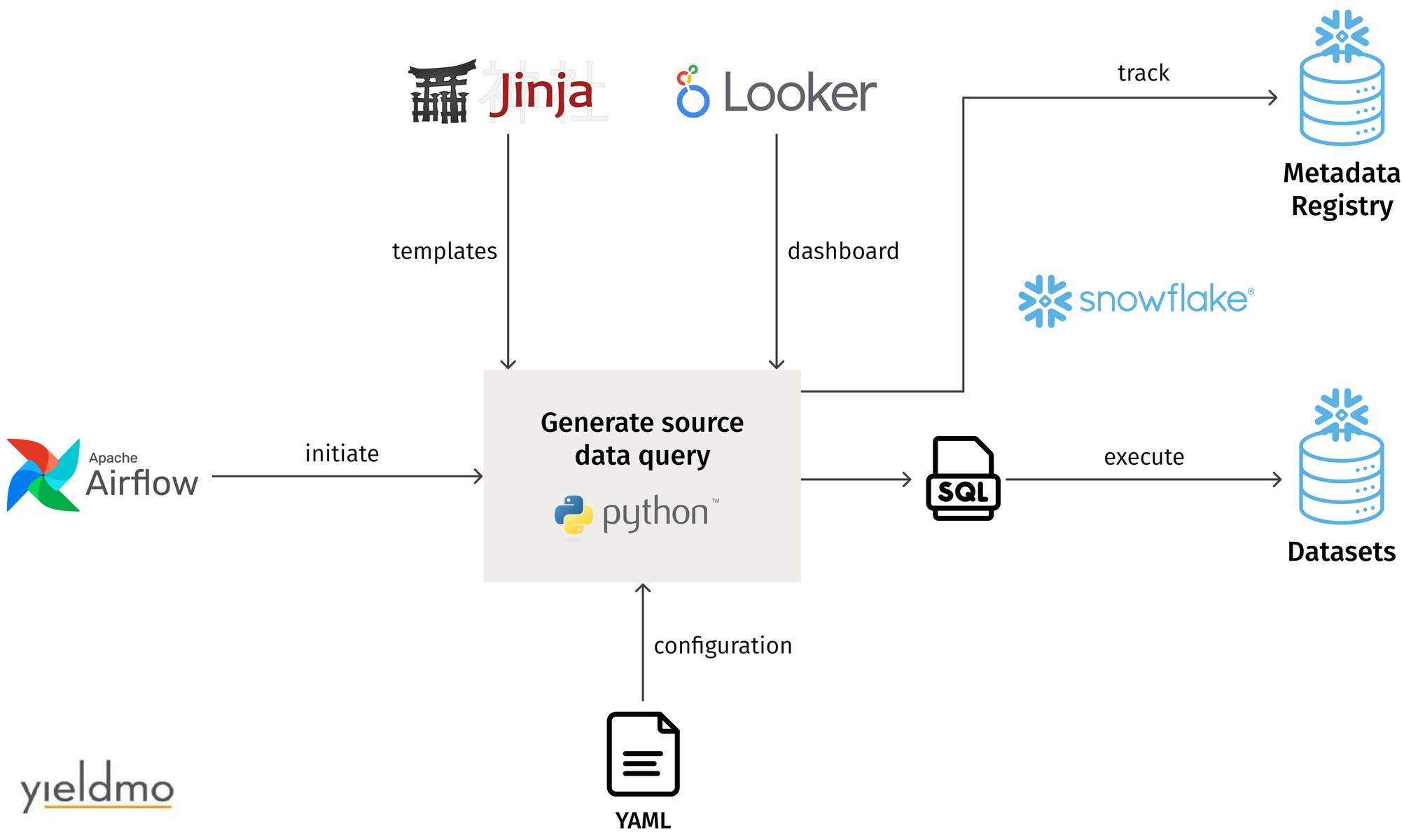
- Configurable, automated ML pipelines that create datasets, train models, deliver artifacts, produce predictions, and deliver predictions to downstream processes.
- A system for standardized definitions of ML model features and data transformations for creating model training and prediction datasets without copying code from one model to another.
- A straightforward way for data scientists to deliver model feature definitions, model configurations, and code that can be directly used in production without having a data engineer rewrite the code for deployment.
- Being able to scale the system to a high volume of predictions at an acceptable time and cost of operation.
- Tracking of the datasets, model artifacts, their versions, dataset and model configuration, and model performance, as well as pipeline monitoring and notifications.

Note that because the system makes predictions in batch, and the ad server caches the results for making ~7M real-time decisions per second, the low latency of the ML model predictions is not a requirement of this ML platform.

Machine learning platform overview

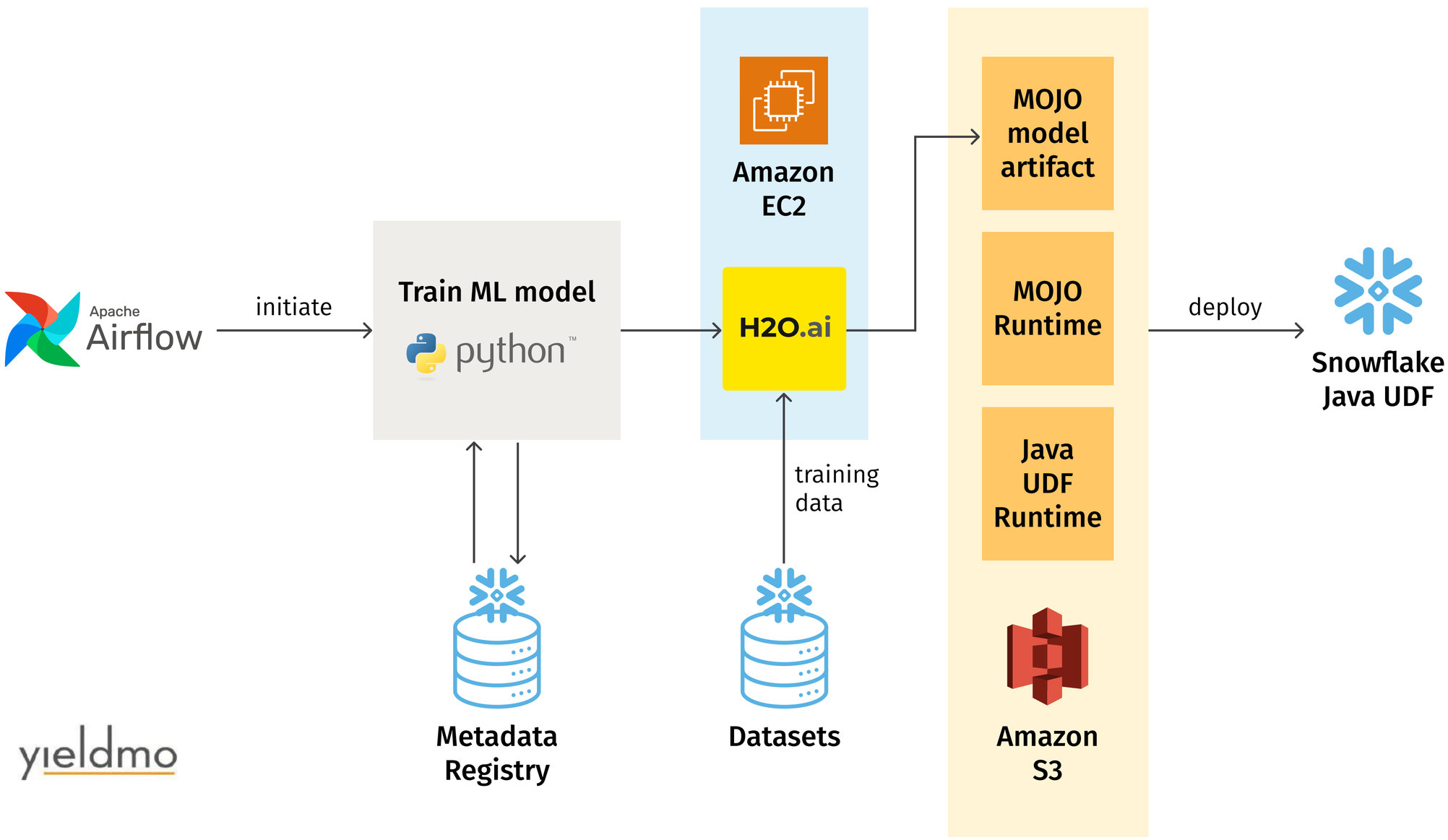
Yieldmo's computing infrastructure is multiple cloud-based. The ML platform described here leverages Snowflake and AWS, but it could similarly use any cloud compute and storage. While most ML model engines can be integrated into the platform by adding an integration module, the production implementation uses Driverless AI from H2O.ai. Two unique features of Driverless AI that help fulfill the platform requirements are the automated feature engineering and model scoring artifacts available as "mojo" packages that contain Java code and model definitions for high-speed and high-throughput scoring.

The ML pipeline journey starts and ends in Yieldmo's Snowflake data warehouse, where all the source data assets, datasets, predictions, and metadata reside. The pipeline is written in Python and uses Jinja2 templates and Looker to generate SQL statements. H2O Driverless AI runs on Amazon EC2 instances configured with or without GPUs, depending on the demands of ML tasks. The pipeline uses Amazon S3 storage as the staging area between H2O and Snowflake. Execution is orchestrated by the Amazon managed workflows for Apache Airflow service (Amazon MWAA).



Dataset Generation

The ML pipeline has two distinct parts, one for model training and the other for batch scoring. The training pipeline generates the training and test datasets, loads them into H2O, trains the model, and produces a trained model artifact. The scoring pipeline can use any previously registered model artifact for the corresponding model type and is configured by default to use the latest accepted one. It produces the dataset for scoring, generates predictions, and performs post-processing of these predictions. The post-processed dataset and its metadata comprise the interface exposed by the ML platform to other systems that consume the predictions, such as the pipelines performing further processing and loading the predictions into the ad server cache. All metadata about the models and datasets are stored in the metadata registry consisting of a set of Snowflake relational database tables.



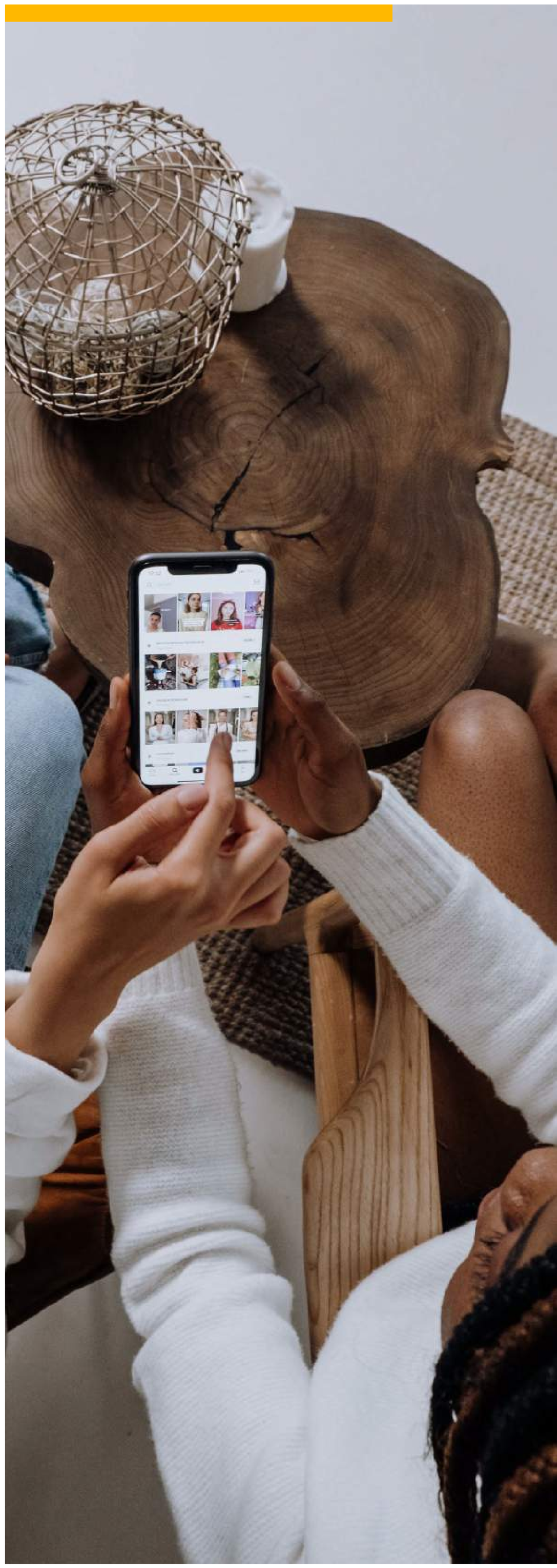
ML Model Training

Standard definitions of model features

Model features in the context of the ML pipeline are the input variables that the ML model uses to predict the output variable, such as the probability of an online user clicking on the ad. These inputs are defined by data scientists and constructed from existing fields in the data warehouse tables. In the simplest case, a feature definition is a mapping of a feature to a column in a specific table. More complex cases perform string manipulation and calculations involving multiple database columns. Without a standard way to define features, data scientists tend to copy SQL code from one model to the next, which is error-prone, hard to maintain, and difficult to reuse and share among team members.

Yieldmo's ML platform leverages Looker to define reusable column mappings and data transformations, and to generate SQL performing only the necessary table joins. Features are exposed via Looker Explores, which data scientists organize into dashboards, picking the features and filters they need in the ML models. Pipelines retrieve the generated SQL code from Looker API.

To accelerate the development cycle and shorten the path to production, the platform also supports feature definitions provided as Jinja2 templates. Data scientists use Jinja2 when they define new features not yet available in Looker, and the models are deployed to production without waiting for the new features to be implemented in Looker. The switch between Jinja2 and Looker templates is very easy due to the configuration-driven architecture of the ML pipelines.



Configuration-driven architecture

One of the design goals of the ML platform was to maximize code reuse and minimize the amount of code that needs to be produced by a data scientist before a model is deployed to production. The platform achieved this goal by making each pipeline component highly configurable with sensible defaults representing the most common use cases. Besides defining and selecting features to be used by the model, which at most requires producing a Jinja2 SQL template, the data scientist needs to create a main configuration file for the model in the YAML format. This file contains the following key items:

- Dataset definitions: lists of features to be used for training, testing, predictions, and post-processing; the SQL template sources for each dataset, such as Looker dashboard names; location of the outputs; other dataset and template configurations, such as sampling rate and other filters.
- ML engine settings: the type of ML engine, target column, and the ML model configuration parameters.

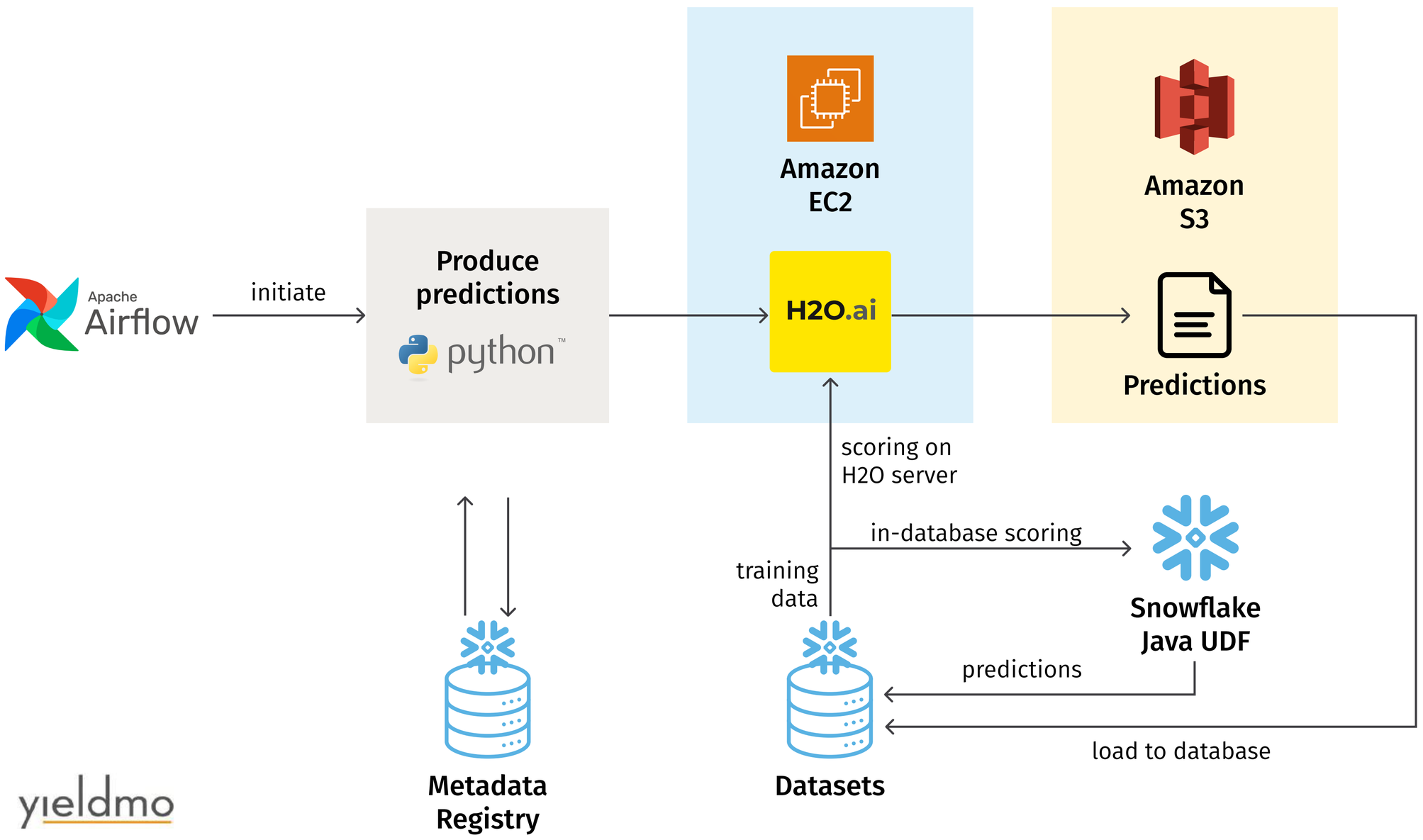
A separate configuration file provides the standard environment settings for development, staging, and production.

The ML pipeline components are organized into Airflow pipelines, with additional business logic, such as checking for the presence of the source data, starting and stopping Amazon EC2 instances as needed, etc., embedded into standard Airflow tasks and configurations. If a new version of the business logic is needed, the data scientists define it using custom Airflow DAGs from the same standard components.

Such an architecture allows for low-code development and deployment of new ML models. In the best-case scenario, all that is needed to put a new model in production is changing the environment setting and adding a schedule and a small model-specific initializer task in the production Airflow environment.

Machine learning inference layer

To support the volume of ads served, batch scoring must be cost-effective. The ML platform supports two types of scoring: one on the H2O Driverless AI server, and the other using Snowflake Java UDFs, which load H2O “mojo” artifact files and perform scoring directly in the data warehouse, leveraging Snowflake’s Snowpark functionality with Java UDFs. The former is used for smaller datasets and during development when it’s easier to load the prediction datasets onto the Driverless AI Amazon EC2 instance. The Snowflake Java UDF scoring is much more cost-effective at scale without having to move the data out of and back into the data warehouse. Once the ML model is trained by the H2O Driverless AI platform, the ML pipeline exports the “mojo” artifact containing the code and model definition data, and loads it into a Snowflake stage. Then, the pipeline defines the UDF, pointing to the model artifact, which the batch scoring pipeline uses to make predictions.



yieldmo

ML Inference

Model metadata registry

When running multiple machine learning models in production, it is crucial for the maintainability of the system to keep track of all the models, their versions, configurations, datasets that were used, and how those datasets were created (aka data lineage). None of the commercially available MLOps systems adequately supported the metadata registry requirements. As a result, Yieldmo's ML platform uses a custom ML metadata registry consisting of relational tables in Snowflake. The main entities of the registry are the "dataset", the "experiment", which is an instance of model training, and the "prediction", which captures the input and output dataset for scoring with a specific experiment. This design closely follows Spark ML's design, with the Datasets, Transformers, and Estimators as the primary entities on which the rest of the Spark ML system is built. The Driverless AI experiments are similar to Estimators, the dataset SQL templates and the batch scoring artifacts are Transformers, and the datasets are, well, Datasets.

Recording the metadata in a relational database registry is incredibly useful for the maintenance and monitoring of the ML platform. For example, it is easy to create a report of a model's performance over time, as well as of the time it took to train the models and to make predictions. The registry supports the reproducibility of datasets and models because it saves the fully rendered SQL queries and exact model configurations at the time the datasets and models are created. The registry also captures the current state of each pipeline execution, which proved to be invaluable for debugging.

Conclusion

Yieldmo's unique technology allows it to collect, organize, analyze, optimize, and store an enormous amount of data at an unsurpassed level of granularity and scale. Taking that massive dataset and making it useful, predictable, and actionable in real time is a complex machine learning problem that only the most sophisticated architectures can solve. Without these tools, a human could not curate inventory or optimize creative performance at the level Yieldmo consistently achieves. The ad experience can significantly impact the success of advertising campaigns, and maximizing that experience drives real outcomes for advertisers. Grid Dynamics assisted in building data platform services, model metadata tracking, and ML platform capabilities at a large scale, helping to reduce time-to-market and support efforts for ML model development.

About Grid Dynamics

Grid Dynamics is a global digital engineering company that co-innovates with the most respected brands in the world to solve complex problems, optimize business operations, and better serve customers. Driven by business impact and agility, we create innovative, end-to-end solutions in digital commerce, AI, data, web UI and UX, and cloud to help clients grow.

Headquartered in Silicon Valley, with delivery centers located throughout the globe, Grid Dynamics is known for architecting revolutionary digital technology platforms for 7 of the 25 largest retailers in the US and 3 of the 10 largest consumer goods companies in the world, as well as leading brands in the digital commerce, manufacturing, finance, healthcare, and high tech sectors.

Our secret sauce? We hire the top 10% of global engineering talent and employ our extensive expertise in emerging technology, lean software development practices, a high-performance product and agile delivery culture, and strategic partnerships with leading technology service providers like Google, Amazon, and Microsoft.

In 2020, Grid Dynamics went public and is trading on the NASDAQ under the GDYN ticker.

About Grid Dynamics

Key facts

- Offices across the US, Mexico, UK, Netherlands, Switzerland, India, and Central and Eastern Europe.
- Thousands of employees across the globe.
- Forrester Leader Midsize Agile Software Development Service Provider Q2 2019.
- Proprietary starter kits developed in collaboration with AWS, Google Cloud, Microsoft Azure, and others.

Areas of expertise

- **Experience engineering**
  Web UI | Mobile | UX | AR/VR
- **Data Science and AI**
  Generative AI | Search | Personalization
  Supply chain | IoT
- **Platform engineering**
  Microservices | MACH | Composable
- **Data engineering**
  Big data | Streaming | MLOps
- **Cloud and DevOps**
  CICD | AIOps | SRE | QE

Clients

	Google	JABIL
align	Levi's	StanleyBlack&Decker
RAYMOND JAMES	fiserv.	AMERICAN EAGLE
Boston Scientific	INSPIRE Brands	HELLO FRESH

Grid Dynamics

trusted engineering partner for digital transformation

Grid Dynamics Holdings, Inc.

5000 Executive Parkway,

Suite 520 / San Ramon, CA

650-523-5000

www.griddynamics.com