ModelOp Center v2.x:
Model Replay

AUGUST 2020

ModelOp

BACKGROUND / USE CASE

Request to "replay a model", including (a) viewing prior scoring runs (b) rerunning the inferences generated at a given date/time in the past for a defined set of transactions or set of transactions. The same input data and exact version of the model should be used.

TABLE OF CONTENTS

Overview

ModelOp Center provides extensive metadata collection and fine-grain versioning and traceability of all models to allow for reproducibility and replayability of each and every version of a model.

A couple of core ModelOp Center elements are foundational to enabling replayability:

- **StoredModel**: Each registered model has a unique GUID associated with the model.
- **DeployableModel:** Once a model has been "submitted" for production, a snapshot of the model--and all of the artifacts associated with the model--is taken and given a unique GUID for the "deployable model" (i.e. a model that is prepped/ready to be deployed). This allows for permanent traceability to each version of the model.
- **DeployedModel**: When a DeployableModel has been physically deployed into the production execution environment, a deployed model record--with a unique deployedModel GUID--is created. This contains reference to metadata such as the timestamp when the model was deployed, the runtime on which it was deployed, and the DeployableModel GUID.
- **REST API**: all of the above objects are available via the ModelOp Center API to allow for reproducible testing, traceable reports, or rerunning a batch of data against the exact version of a model that was deployed at a given timestamp in the past.

Steps to Enable in ModelOp Center

The below highlights the steps to execute the test cases as defined above in ModelOp Center.

For Online Models

1. The system that is making inference requests should save the transaction_id, timestamp and model endpoint (url or queue) in a database table.
2. To replay a model inference, acquire the timestamp and model endpoint for a particular transaction from the aforementioned table.
3. Extract the engine name from the model endpoint URL.
4. Make an api call to ModelOp Center with the timestamp and engine name. This will return a deployed_model_id and the associated deployable _model_id.
5. Go to the ModelOp Center UI and find the model version corresponding to the deployable_model_id.
6. Create a batch job, upload a data file containing the data for the desired transaction. The output file will contain the inference result.

For Batch Models:

- Batch jobs must be Initiated by an external process using the moc cli or the camunda java sdk both of which will return a batch_id.
- The batch id should be stored in a database along with the location of the output file when the batch job completes.
- Another process (or the same one) should write out a transaction_id, batch_job_id pair for every record in the output file.
- To replay a model inference for a particular transaction_id, acquire the batch_job_id from the database table described in step 3.

- Make an api call to MOC using the batch_job_id to get the deployed_model_id and the associated deployable _model_id.
- Go to MOC UI and find the model version corresponding to the deployable_model_id.
- Create a batch job, upload a data file containing the data for the desired transaction. The output file will contain the inference result.

Looking Ahead

The ModelOp Center roadmap is guided by our vision but refined by our customers' needs. Combined with a world-class engineering team, we produce feature releases rapidly to continue to be the market leaders in the ModelOps space.

Related to these test cases, while ModelOp Center already has the comprehensive data model to address the stated test cases (per above), the ModelOp Center roadmap has several UX/UI updates, including:

- **Enabling Search within the Jobs Page:** within the Jobs page, enabling search by defined parameters
- **Viewing prior batch runs within Model Version Page:** while all prior Scoring job runs are currently available in the Jobs page, a view will be added to view the prior batch scoring runs within the model details page to allow for quick access
- **Search by Custom Metadata:** Within the UI, ability to search by custom metadata, such as finding a version by deployment date, approval date, risk level, or other custom metadata
- **Auto-prepare a Model Replay:** Within the UI, ability to select a version of a model and automatically prepare a batch training or scoring job with the selected version of the model and the training/test data set associated with that specific version (DeployableModel)