Red Hat OpenShift AI Self-Managed 2.5

Monitoring data science models
Red Hat OpenShift AI Self-Managed 2.5 Monitoring data science models
Abstract

The TrustyAI service provides data scientists with value-added capabilities, such as explainability (enriching model execution information through XAI algorithms) and model bias detection.
# Table of Contents

CHAPTER 1. ENABLING THE TRUSTYAI SERVICE FOR A DATA SCIENCE PROJECT ............................................. 3

CHAPTER 2. MONITORING MODELS FOR BIAS ......................................................................................... 6
  2.1. CONFIGURING BIAS MONITORING FOR A MODEL BY USING THE CLI ................................. 6
  2.2. VIEWING BIAS METRICS FOR A MODEL .................................................................................. 8
  2.3. SUPPORTED BIAS METRICS ................................................................................................. 8
When you install OpenShift AI, the TrustyAI Operator is included with the other components in the OpenShift AI namespace. The TrustyAI operator is responsible for deploying and managing TrustyAI services, as well as managing all other resources required by TrustyAI.

These managing tasks include configuring storage, creating the service monitors, and configuring the serving runtimes and routes.

The TrustyAI operator manages all enabled TrustyAI services across any number of projects on your cluster.

For each data science project (namespace) that contains models for which you want to monitor bias metrics, enable an instance of the TrustyAI service.

**NOTE:** You should enable only one instance of the TrustyAI service in a project. The presence of multiple instances in the same project can result in unexpected behavior.

**Prerequisites**

- On the OpenShift cluster where OpenShift AI is installed, you have enabled user workload monitoring as described in Enabling monitoring for user-defined projects.
- You have installed OpenShift AI as described in Installing the Red Hat OpenShift Data Science Operator.
- The **trustyai** component is set to **Managed** for the Red Hat OpenShift Data Science Operator. To verify this setting, navigate to Operators → Installed Operators → Red Hat OpenShift Data Science Operator → Data Science Cluster. Select the **default** instance and then click **YAML**. Scroll down to view the `spec:components` setting:

  ```yaml
  trustyai:
    devFlags: {}
    managementState: Managed
  ```

  **NOTE:** If the **trustyai** component is set to **Removed**, edit the YAML file to set it to **Managed**.
- You have logged in to OpenShift AI.
- If you are using specialized OpenShift AI groups, you are part of the user group or admin group (for example, **oai-users** or **oai-admins**) in OpenShift.
You have created a data science project, as described in Creating a data science project, that contains (or will contain) the models that you want to monitor.

Procedure
Add an instance of the TrustyAI service to each data science project that contains models that you want to monitor. The TrustyAI service instance provides the URL that a developer uses to monitor and analyze any number of models deployed into a data science project.

To enable the TrustyAI service from the command line, follow these steps:

1. Login to your cluster.
   ```bash
   oc login
   ```
2. Navigate to the data science project that contains (or will contain) the models that you want to monitor.
   ```bash
   oc project <project-name>
   ```
   For example:
   ```bash
   oc project mydsproject
   ```
3. Create a TrustyAIService custom resource (CR) file, for example trustyai_crd.yaml:
   ```yaml
   apiVersion: trustyai.opendatahub.io.trustyai.opendatahub.io/v1alpha1
   kind: TrustyAIService
   metadata:
     name: trustyai-service
   spec:
     storage:
       format: "PVC"
       folder: "/inputs"
       size: "1Gi"
     data:
       filename: "data.csv"
       format: "CSV"
     metrics:
       schedule: "5s"
       batchSize: 5000 # Optional, default is 5000
   ```

Here is a description of the fields:

**metadata.name**
- The name of the TrustyAI service instance.

**spec.storage.format**
- The storage format for the data. Currently, only persistent volume control (PVC) is supported.

**spec.storage.folder**
- The location within the PVC where the data will be stored.

**spec.storage.size**
- The size of the PVC to request.
spec.data.filename
The suffix for the stored data files.

spec.data.format
The format of the data. Currently, only comma-separated value (CSV) format is supported.

spec.metrics.schedule
The interval at which the metrics are calculated. The default is 5s. The duration is specified with the ISO-8601 format. For example, 5s for 5 seconds, 5m for 5 minutes, and 5h for 5 hours.

spec.metrics.batchSize
The observation’s historical window size to use for metrics calculation. The default is 5000 (that is, the metrics are calculated by using the latest 5000 inferences).

4. Add the TrustyAI service’s CR to your project:

   oc apply -f trustyai_crd.yaml

   This command should return output similar to the following:

   trusty-service created

Verification
To verify that you enabled the TrustyAI Service:

   oc get pods | grep trustyai

   You should see a response similar to the following:

   trustyai-service-5d45b5884f-96h5z 1/1 Running
CHAPTER 2. MONITORING MODELS FOR BIAS

IMPORTANT

The TrustyAI features are for Technology Preview only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs), might not be functionally complete, and Red Hat does not recommend using them for production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process. For more information on Red Hat Technology Preview features, see Technology Preview Features Scope.

Note that TrustyAI features are supported only with ModelMesh serving.

As a data scientist, you might need to monitor your machine learning models for bias. This means monitoring for algorithmic deficiencies that might skew the outcomes or decisions that the model produces. Importantly, this type of monitoring helps you to ensure that the model is not biased against particular protected groups or features.

Red Hat OpenShift AI provides a set of metrics that help you to monitor your models for bias. You can use the OpenShift AI interface to choose an available metric and then configure model-specific details such as a protected attribute, the privileged and unprivileged groups, the outcome you want to monitor, and a threshold for bias. You then see a chart of the calculated values for a specified number of model inferences.

The sections that follow describe how to configure your models for bias monitoring and how to view and interpret the resulting metrics.

2.1. CONFIGURING BIAS MONITORING FOR A MODEL BY USING THE CLI

IMPORTANT

The TrustyAI features are for Technology Preview only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs), might not be functionally complete, and Red Hat does not recommend using them for production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process. For more information on Red Hat Technology Preview features, see Technology Preview Features Scope.

To monitor a deployed model for bias, you must first configure bias metrics. When you configure bias metrics, you specify details relevant to your model. For example, details such as a protected attribute, privileged and unprivileged groups, a model outcome and value that you want to monitor, and the acceptable threshold for bias.

Prerequisites

- You are familiar with the bias metrics that OpenShift AI supports and how to interpret them.
- Your OpenShift cluster administrator has installed OpenShift AI and deployed the TrustyAI service for the data science project where the models are deployed.
You installed the OpenShift command line interface (oc) as described in Get Started with the CLI.

Procedure

1. In a terminal window, log into the OpenShift cluster where OpenShift AI is deployed.
   
   oc login

2. Set the TRUSTY_ROUTE variable to the external route for the TrustyAI service pod.
   
   $TRUSTY_ROUTE=http://$(oc get route/trustyai-service --template={{.spec.host}})

3. Optionally, get the full list of TrustyAI service endpoints and payloads.
   
   curl --location $TRUSTY_ROUTE/q/openapi

4. Configure bias metrics by running a command that uses the following syntax and payload structure:

   Syntax:

   ```
curl -sk  -X POST --location $TRUSTY_ROUTE/metrics/spd/request \
   --header 'Content-Type: application/json' \
   --data <payload>
```

   Payload structure:

   - **modelId**: The name of the model to query.
   - **protectedAttribute**: The name of the feature that distinguishes the groups that you are checking for fairness.
   - **privilegedAttribute**: The suspected favored (positively biased) class.
   - **unprivilegedAttribute**: The suspected unfavored (negatively biased) class.
   - **outcomeName**: The name of the output that provides the output you are examining for fairness.
   - **favorableOutcome**: The value of the outcomeName output that describes the favorable or desired model prediction.
   - **batchSize**: The number of previous inferences to include in the calculation.

For example:

```bash
curl -sk  -X POST --location $TRUSTY_ROUTE/metrics/group/fairness/spd/ \
   --header 'Content-Type: application/json' \
   --data "{" \
   "modelId": "demo-loan-nn-onnx-alpha",
```


```json
"protectedAttribute": "Is Male-Identifying?",
"privilegedAttribute": 1.0,
"unprivilegedAttribute": 0.0,
"outcomeName": "Will Default?",
"favorableOutcome": 0,
"batchSize": 5000
}
```

**Verification**

The bias metrics request should return output similar to the following:

```json
{
  "timestamp": "2023-10-24T12:06:04.586+00:00",
  "type": "metric",
  "value": -0.0029676404469311524,
  "namedValues": null,
  "specificDefinition": "The SPD of -0.002968 indicates that the likelihood of Group: Is Male-Identifying?=1.0 receiving Outcome: Will Default?=0 was -0.296764 percentage points lower than that of Group: Is Male-Identifying?=0.0.",
  "name": "SPD",
  "id": "d2707d5b-cae9-41aa-bcd3-d950176cbbaf",
  "thresholds": {"lowerBound": -0.1,"upperBound": 0.1,"outsideBounds": false}
}
```

The **specificDefinition** field is useful in understanding the real-world interpretation of these metric values. For this example, the model is quite fair over the **Is Male-Identifying?** field, with the rate of positive outcome only differing by approximately -0.3%.

### 2.2. VIEWING BIAS METRICS FOR A MODEL

When you configure your model for bias monitoring, you can view the metrics that you configured.

**Prerequisites**

- You configured bias metrics for your model as described in Configuring bias monitoring for a model.
- You have admin access to the OpenShift web console.

**Procedure**

- From the Administrator perspective within the OpenShift web console, select **Observe → Metrics**.

### 2.3. SUPPORTED BIAS METRICS

Red Hat OpenShift AI supports the following bias metrics.

**Statistical Parity Difference**

*Statistical Parity Difference* (SPD) is the difference in the probability of a favorable outcome prediction between unprivileged and privileged groups. The formal definition of SPD is the following:

\[
SPD = p(\hat{y} = 1 \mid D_u) - p(\hat{y} = 1 \mid D_p)
\]
\[ \hat{y} = 1 \] is the favorable outcome.

\[ D_u \] and \[ D_p \] are the unprivileged and privileged group data.

You can interpret SPD values as follows:

- A value of 0 means that the model is behaving fairly for a selected attribute (for example, race, gender).
- A value in the range \(-0.1\) to \(0.1\) means that the model is reasonably fair for a selected attribute. Instead, you can attribute the difference in probability to other factors, such as the sample size.
- A value outside the range \(-0.1\) to \(0.1\) indicates that the model is unfair for a selected attribute.
- A negative value indicates that the model has bias against the unprivileged group.
- A positive value indicates that the model has bias against the privileged group.

**Disparate Impact Ratio**

Disparate Impact Ratio (DIR) is the ratio of the probability of a favorable outcome prediction for unprivileged groups to that of privileged groups. The formal definition of DIR is the following:

\[
\text{DIR} = \frac{p(\hat{y} = 1 \mid D_u)}{p(\hat{y} = 1 \mid D_p)}
\]

- \( \hat{y} = 1 \) is the favorable outcome.
- \( D_u \) and \( D_p \) are the unprivileged and privileged group data.

The threshold to identify bias depends on your own criteria and specific use case.

For example, if your threshold for identifying bias is represented by a DIR value below 0.8 or above 1.2, you can interpret the DIR values as follows:

- A value of 1 means that the model is fair for a selected attribute.
- A value of between 0.8 and 1.2 means that the model is reasonably fair for a selected attribute.
- A value below 0.8 or above 1.2 indicates bias.