Red Hat OpenShift AI Self-Managed 2.6

Monitoring data science models

Legal Notice

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

PREFACE

NOTE

The monitoring models for bias (TrustyAI feature) is supported only on a *multi-model serving platform* that is based on the ModelMesh component. It is not supported on a *single model serving platform* that is based on the KServe component.

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

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 management 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 the data scientists on your team want to monitor bias metrics, enable an instance of the TrustyAI service. You can enable an instance of the TrustyAI service by using the OpenShift AI dashboard or by using the OpenShift command-line interface (CLI).

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.

1.1. ENABLING THE TRUSTYAI SERVICE BY USING THE DASHBOARD

Enable an instance of the TrustyAI service on each data science project that contains models that the data scientist wants to monitor. The TrustyAI service instance provides the URL that the data scientist uses to monitor and analyze any number of models deployed into a data science project.

The following procedure describes how to use the OpenShift AI dashboard to enable an instance of the TrustyAI service.

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 enabled the multi-model serving platform.

- You have enabled the multi-model serving platform as described in Enabling the multi-model serving platform.

NOTE

The monitoring models for bias (TrustyAI feature) is supported only on a *multi-model serving platform* that is based on the ModelMesh component. It is not supported on a *single model serving platform* that is based on the KServe component.

- You have installed OpenShift AI as described in Installing the Red Hat OpenShift AI Operator.

- The **trustyai** component is set to **Managed** for the Red Hat OpenShift AI Operator. To verify this setting, navigate to **Operators → Installed Operators → Red Hat OpenShift AI Operator → Data Science Cluster**. Select the **default** instance and then click **YAML**. Scroll down to view the **spec:components** setting:

```
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, **rhoai-users** or **rhoai-admins**) in OpenShift.

- The data scientist has created a data science project, as described in Creating a data science project, that contains (or will contain) the models that the data scientist wants to monitor.

Procedure

1. Log in to OpenShift AI.

2. From the OpenShift AI dashboard, click **Data Science Projects**.

3. In the **Data Science Projects** page, click the name of the project that contains (or will contain) the models that the data scientist wants to monitor.

4. In the project details, page, click **Settings**.

5. Select the **Enable TrustyAI** checkbox.

Verification

1. In the OpenShift Container Platform web console, click **Workloads → Pods**.

2. From the project list, select the project namespace in which you enabled TrustyAI.

3. Confirm that there is a running pod for the TrustyAI service. The pod has a naming pattern similar to the following example:

```
trustyai-service-5d45b5884f-96h5z
```

1.2. ENABLING THE TRUSTYAI SERVICE BY USING THE CLI

Enable an instance of the TrustyAI service on each data science project that contains models that the data scientist wants to monitor. The TrustyAI service instance provides the URL that a data scientist uses to monitor and analyze any number of models deployed into a data science project.

The following procedure describes how to use the OpenShift command-line interface (CLI) to enable an instance of the TrustyAI service.

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 enabled the multi-model serving platform.

- You have enabled the multi-model serving platform as described in Enabling the multi-model serving platform.

NOTE

The monitoring models for bias (TrustyAI feature) is supported only on a *multi-model serving platform* that is based on the ModelMesh component. It is not supported on a *single model serving platform* that is based on the KServe component.

- You have installed OpenShift AI as described in Installing the Red Hat OpenShift AI Operator.

- The **trustyai** component is set to **Managed** for the Red Hat OpenShift AI Operator.
  To verify this setting, navigate to **Operators → Installed Operators → Red Hat OpenShift AI Operator → Data Science Cluster**. Select the **default** instance and then click **YAML**. Scroll down to view the **spec:components** setting:

  ```
  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, **rhoai-users** or **rhoai-admins**) in OpenShift.

- The data scientist has created a data science project, as described in Creating a data science project, that contains (or will contain) the models that the data scientist wants to monitor.

Procedure

1. Login to your cluster.

   ```
   oc login
   ```

2. Navigate to the data science project that contains (or will contain) the models that the data scientist wants to monitor.

   ```
   oc project <project-name>
   ```

   For example:

   ```
   oc project mydsproject
   ```

3. Create a **TrustyAIService** custom resource (CR) file, for example **trustyai_crd.yaml**:

   ```
   apiVersion: 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) format is supported.

spec.storage.folder

The location within the PVC where you want to store the data.

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 to calculate the metrics. 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 returns output similar to the following:

```
trusty-service created
```

Verification

Use this command 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. CONFIGURING BIAS METRICS FOR A MODEL

As a data scientist, you might want 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.

2.1. CREATING A BIAS METRIC

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

You can create a bias metric for a model by using the OpenShift AI dashboard or by using the OpenShift command-line interface (CLI).

2.1.1. Creating a bias metric by using the dashboard

The following procedure describes how to use the OpenShift AI dashboard to create a bias metric for a model.

Prerequisites

- You are familiar with the bias metrics that OpenShift AI supports and how to interpret them.

- You are familiar with the specific data set schema and understand the names and meanings of the inputs and outputs.

- Your OpenShift cluster administrator has installed OpenShift AI and enabled the TrustyAI service for the data science project where the models are deployed.

- You have logged in to Red Hat OpenShift AI.

Procedure

1. In the left menu of the OpenShift AI dashboard, click **Model Serving**.

2. On the **Model serving** page, select your project from the drop-down list.

3. Click the name of the model that you want to configure bias metrics for.

4. In the metrics page for the model, click the **Model bias** tab.

5. Click **Configure**.

6. In the **Configure bias metrics** dialog, complete the following steps to configure bias metrics:

a. In the **Metric name** field, type a unique name for your bias metric. Note that you cannot change the name of this metric later.

b. From the **Metric type** list, select one of the metrics types that are available in OpenShift AI.

c. In the **Protected attribute** field, type the name of an attribute in your model that you want to monitor for bias.

TIP

You can use a **curl** command to query the metadata endpoint and view input attribute names and values. For example: **curl -H "Authorization: Bearer $TOKEN" $TRUSTY_ROUTE/info | jq ".[0].data.inputSchema"**

d. In the **Privileged value** field, type the name of a privileged group for the protected attribute that you specified.

e. In the **Unprivileged value** field, type the name of an unprivileged group for the protected attribute that you specified.

f. In the **Output** field, type the name of the model outcome that you want to monitor for bias.

TIP

You can use a **curl** command to query the metadata endpoint and view output attribute names and values. For example: **curl -H "Authorization: Bearer $TOKEN" $TRUSTY_ROUTE/info | jq ".[0].data.outputSchema"**

g. In the **Output value** field, type the value of the outcome that you want to monitor for bias.

h. In the **Violation threshold** field, type the bias threshold for your selected metric type. This threshold value defines how far the specified metric can be from the fairness value for your metric, before the model is considered biased.

i. In the **Metric batch size** field, type the number of model inferences that OpenShift AI includes each time it calculates the metric.

7. Ensure that the values you entered are correct.
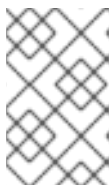
NOTE

You cannot edit a model bias metric configuration after you create it. Instead, you can duplicate a metric and then edit (configure) it; however, the history of the original metric is not applied to the copy.

8. Click **Configure**.

Verification

- The **Bias metric configuration** page shows the bias metrics that you configured for your model.

Next step

To view metrics, on the **Bias metric configuration** page, click **View metrics** in the upper-right corner. For more information, see Viewing bias metrics for a model .

2.1.2. Creating a bias metric by using the CLI

The following procedure describes how to use the OpenShift command-line interface (CLI) to create a bias metric for a model.

Prerequisites

- You are familiar with the bias metrics that OpenShift AI supports and how to interpret them.

- You are familiar with the specific data set schema and understand the names and meanings of the inputs and outputs.

- Your OpenShift cluster administrator has installed OpenShift AI and enabled 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.

- You have a user token for authentication or a token from a service account with sufficient privileges. For example, you can use the following command to get the token on OpenShift Container Platform:

  ```
  TOKEN=$(oc whoami -t)
  ```

Procedure

1. In a terminal window, log in to 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 -H "Authorization: Bearer $TOKEN" --location $TRUSTY_ROUTE/q/openapi
   ```

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

   ```
   curl -sk -H "Authorization: Bearer $TOKEN" -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:

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

```
{
  "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. DUPLICATING A BIAS METRIC

If you want to edit an existing metric, you can duplicate (copy) it in the OpenShift AI interface and then edit the values in the copy. However, note that the history of the original metric is not applied to the copy.

Prerequisites

- You are familiar with the bias metrics that OpenShift AI supports and how to interpret them.

- You are familiar with the specific data set schema and understand the names and meanings of the inputs and outputs.

- You have logged in to Red Hat OpenShift AI.

- There is an existing bias metric that you want to duplicate.

Procedure

1. In the left menu of the OpenShift AI dashboard, click **Model Serving**.

2. On the **Model serving** page, click the name of the model with the bias metric that you want to duplicate.

3. In the metrics page for the model, click the **Model bias** tab.

4. Click the action menu (⋮) next to the metric that you want to copy and then click **Duplicate**.

5. In the **Configure bias metrics** dialog, follow these steps:

    a. In the **Metric name** field, type a unique name for your bias metric. Note that you cannot change the name of this metric later.

    b. Change the values of the fields as needed. For a description of these fields, see Creating a bias metric by using the dashboard.

6. Ensure that the values you entered are correct, and then click **Configure**.

Verification

- The **Bias metric configuration** page shows the bias metrics that you configured for your model.

Next step

To view metrics, on the **Bias metric configuration** page, click **View metrics** in the upper-right corner. For more information, see Viewing bias metrics for a model .

2.3. DELETING A BIAS METRIC

You can delete a bias metric for a model by using the OpenShift AI dashboard or by using the OpenShift command-line interface (CLI).

2.3.1. Deleting a bias metric by using the dashboard

The following procedure describes how to use the OpenShift AI dashboard to delete a bias metric for a model.
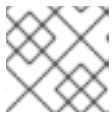
Prerequisites

- You have logged in to Red Hat OpenShift AI.

- There is an existing bias metric that you want to delete.

Procedure

1. In the left menu of the OpenShift AI dashboard, click **Model Serving**.

2. On the **Model serving** page, click the name of the model with the bias metric that you want to delete.

3. In the metric page for the model, click the **Model bias** tab.

4. Click the action menu (⋮) next to the metric that you want to delete and then click **Delete**.

5. In the **Delete bias metric** dialog, type the metric name to confirm the deletion.

> **NOTE**
>
> You cannot undo deleting a bias metric.

6. Click **Delete bias metric**.

Verification

- The **Bias metric configuration** page does not show the bias metric that you deleted.

2.3.2. Deleting a bias metric by using the CLI

The following procedure describes how to use the OpenShift command-line interface (CLI) to delete a bias metric for a model.

Prerequisites

- You have logged in to Red Hat OpenShift AI.

- There is an existing bias metric that you want to delete.

- You installed the OpenShift command line interface (**oc**) as described in Get Started with the CLI.

- You have a user token for authentication or a token from a service account with sufficient privileges. For example, you can use the following command to get the token on OpenShift Container Platform:

    ```
    TOKEN=$(oc whoami -t)
    ```

Procedure

1. In a terminal window, log in to the OpenShift cluster where OpenShift AI is deployed.

    ```
    oc login
    ```

2. Optional: To list all currently active requests for a metric, use **GET /metrics/{{metric}}/requests**. For example, to list all currently scheduled SPD metrics, type:

```
curl -H "Authorization: Bearer $TOKEN" -X GET --location
"$TRUSTY_ROUTE/metrics/spd/requests"
```

3. To delete a metric, send an HTTP **DELETE** request to the **/metrics/$METRIC/request** endpoint to stop the periodic calculation, including the id of periodic task that you want to cancel in the payload. For example:

```
curl -H "Authorization: Bearer $TOKEN" -X DELETE --location
"$TRUSTY_ROUTE/metrics/spd/request" \
    -H "Content-Type: application/json" \
    -d "{
        \"requestId\": \"3281c891-e2a5-4eb3-b05d-7f3831acbb56\"
      }"
```

Verification

Use **GET /metrics/{{metric}}/requests** to list all currently active requests for the metric and verify the metric that you deleted is not shown. For example:

```
curl -H "Authorization: Bearer $TOKEN" -X GET --location "$TRUSTY_ROUTE/metrics/spd/requests"
```

CHAPTER 3. VIEWING BIAS METRICS FOR A MODEL

After you configure your model for bias monitoring, you can use the OpenShift AI dashboard to view and update the metrics that you configured.

Prerequisite

- You configured bias metrics for your model as described in Creating a bias metric .

Procedure

1. In the OpenShift AI dashboard, click **Model Serving**.

2. On the **Model serving** page, click the name of a model that you want to view bias metrics for.

3. To update the metrics shown on the page, follow these steps:

   a. In the **Metrics to display** section, use the **Select a metric** list to select a metric to show on the page.

   > **NOTE**
   >
   > Each time you select a metric to show on the page, an additional **Select a metric** list appears. This enables you to show multiple metrics on the page.

   b. From the **Time range** list in the upper-right corner, select a value.

   c. From the **Refresh interval** list in the upper-right corner, select a value.
   The metrics page shows the metrics that you selected.

4. Optional: To remove one or more metrics from the page, in the **Metrics to display** section, perform one of the following actions:

   - To remove an individual metric, click the cancel icon (✖) next to the metric name.

   - To remove all metrics, click the cancel icon (✖) in the **Select a metric** list.

5. Optional: To return to configuring bias metrics for the model, on the metrics page, click **Configure** in the upper-right corner.

Verification

- The metrics page shows the metrics selections that you made.

CHAPTER 4. 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:

$$DIR = \frac{p(\hat{y} = 1 \mid D_u)}{p(\hat{y} = 1 \mid D_p)}$$

- $\hat{y}$ = 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.