# Responsible AI Model with Azure ML Studio

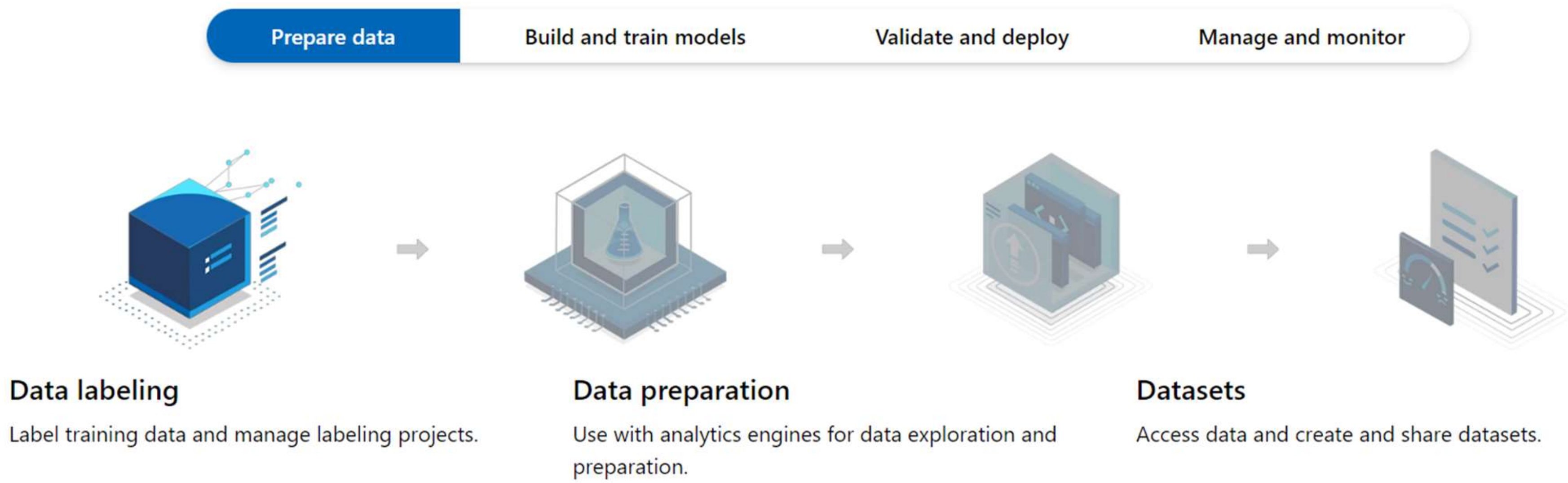
Sarbani Maiti

VP Data Science  at Veersa technology

Email: sarbaniiitb2020@gmail.com

28-Jun-2023

# Azure ML – Supports end to end ML Lifecycle



# Azure ML – Supports end to end ML Lifecycle

Prepare data | **Build and train models** | Validate and deploy | Manage and monitor

### Notebooks
Use collaborative Jupyter notebooks with attached compute.

### Automated machine learning
Automatically train and tune accurate models.

### Drag-and-drop designer
Design with a drag-and-drop development interface.

### Experiments
Run experiments and create and share custom dashboards.

### CLI and Python SDK
Accelerate the model training process while scaling up and out on Azure compute.

### Visual Studio Code and GitHub
Use familiar tools and switch easily from local to cloud training.

### Compute instance
Develop in a managed and secure environment with dynamically scalable CPUs, GPUs, and supercomputing clusters.

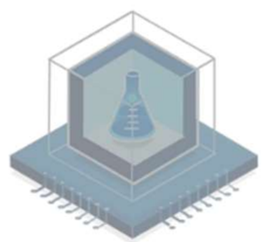
### Open-source libraries and frameworks
Get built-in support for Scikit-learn, PyTorch, TensorFlow, Keras, Ray RLLib, and more.

# Azure ML – Supports end to end ML Lifecycle

Prepare data | Build and train models | **Validate and deploy** | Manage and monitor

### Managed endpoints

Deploy models for batch and real-time inference quickly and easily.



### Pipelines and CI/CD

Automate machine learning workflows.

### Prebuilt images

Access container images with frameworks and libraries for inference.

### Model repository

Share and track models and data.

### Hybrid and multicloud

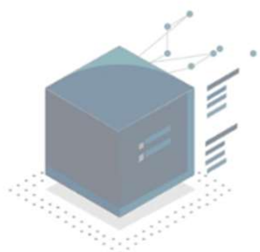
Train and deploy models on premises and across multicloud environments.

### Optimize models

Accelerate training and inference and lower costs with ONNX Runtime.

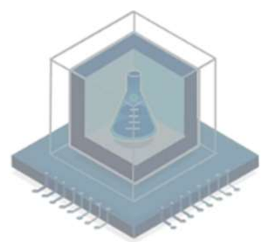
# Azure ML – Supports end to end ML Lifecycle

Prepare data | Build and train models | Validate and deploy | **Manage and monitor**



### Monitoring and analysis

Track, log, and analyze data, models, and resources.



### Data drift

Detect drift and maintain model accuracy.

### Error analysis

Debug models and optimize model accuracy.

### Auditing

Trace machine learning artifacts for compliance.

### Policies

Use built-in and custom policies for compliance management.

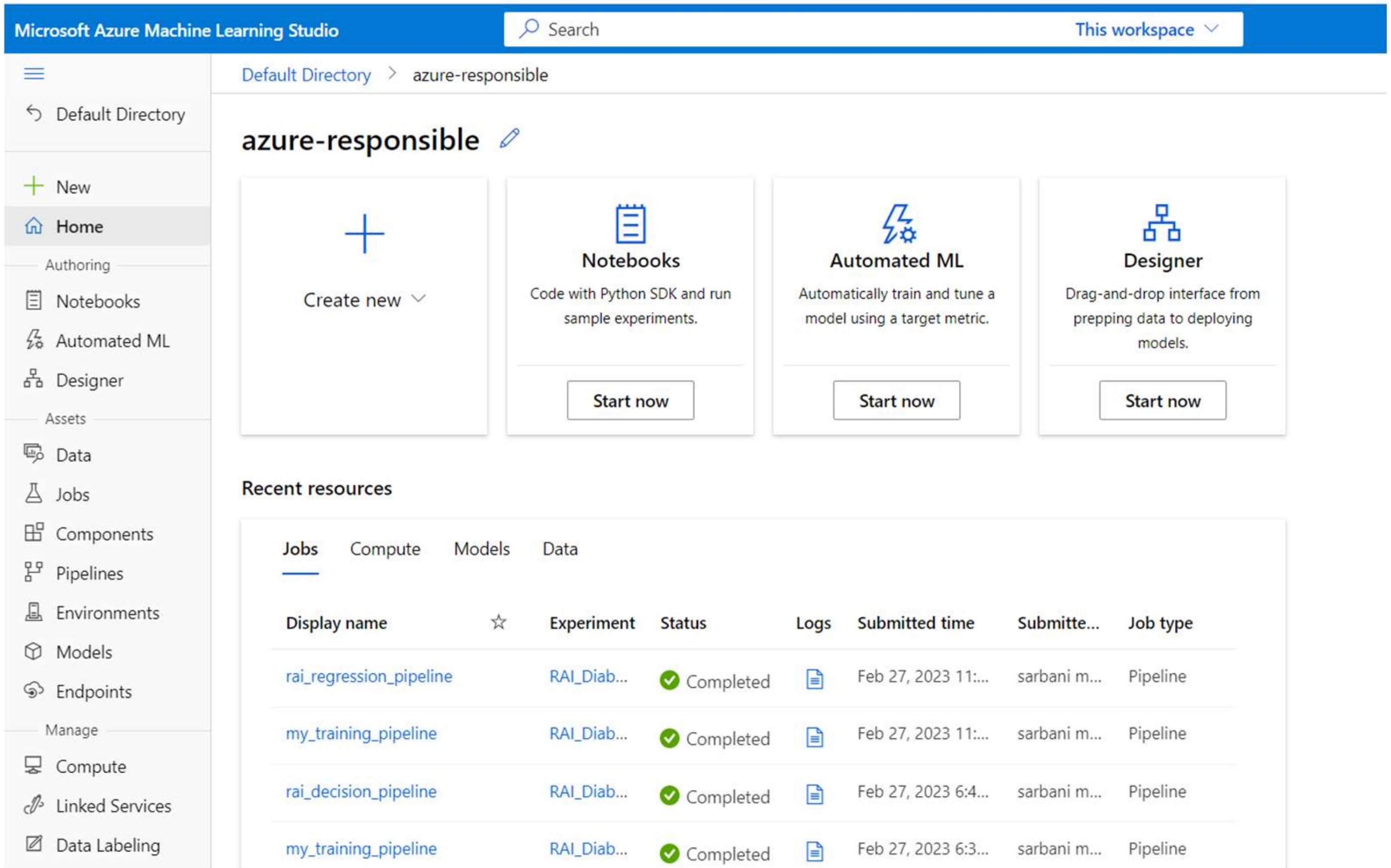
### Security

Enjoy continuous monitoring with Azure Security Center.

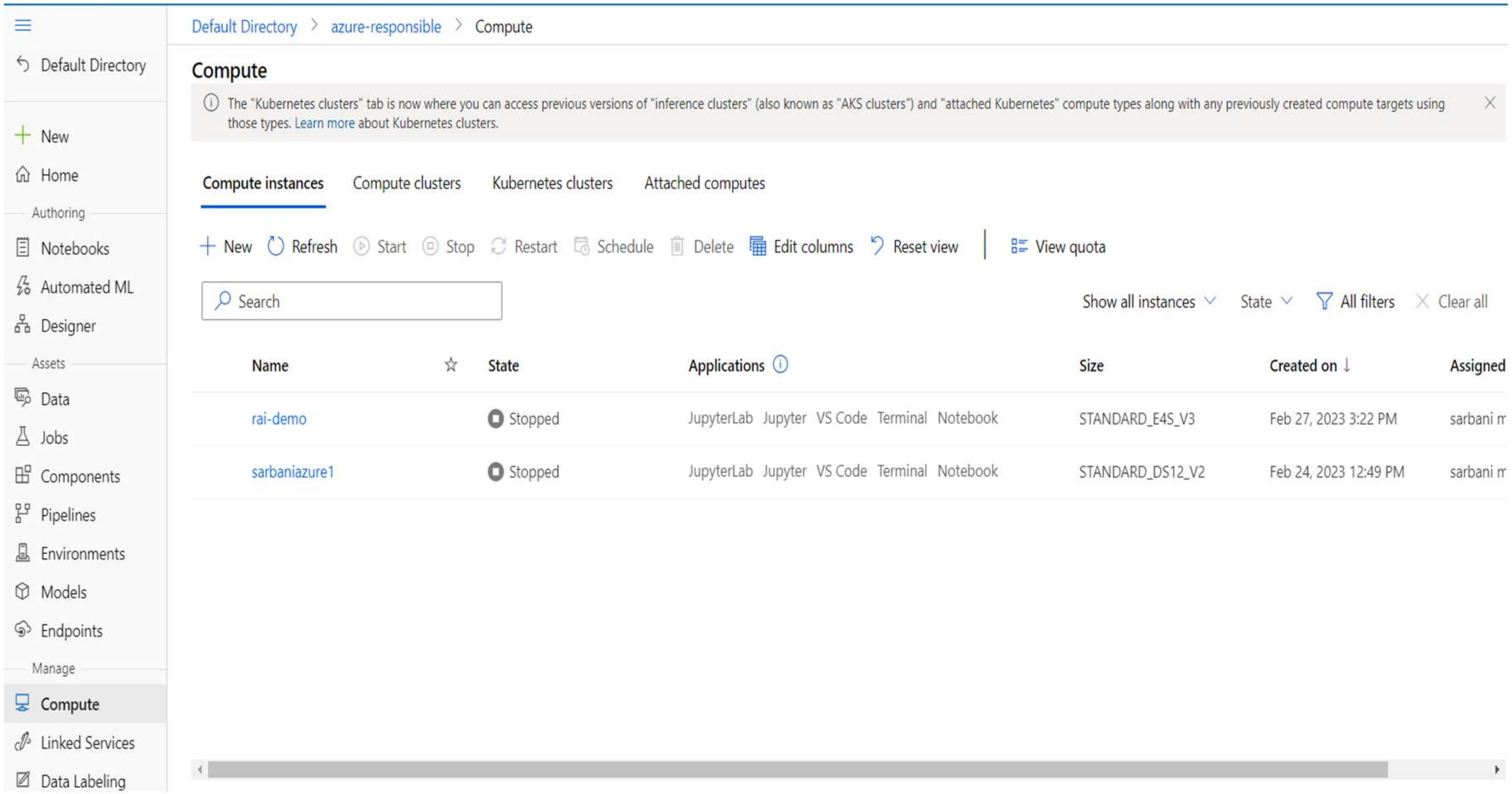
### Cost control

Apply quota management and automatic shutdown.

# Azure ML Studio



# Azure ML Studio

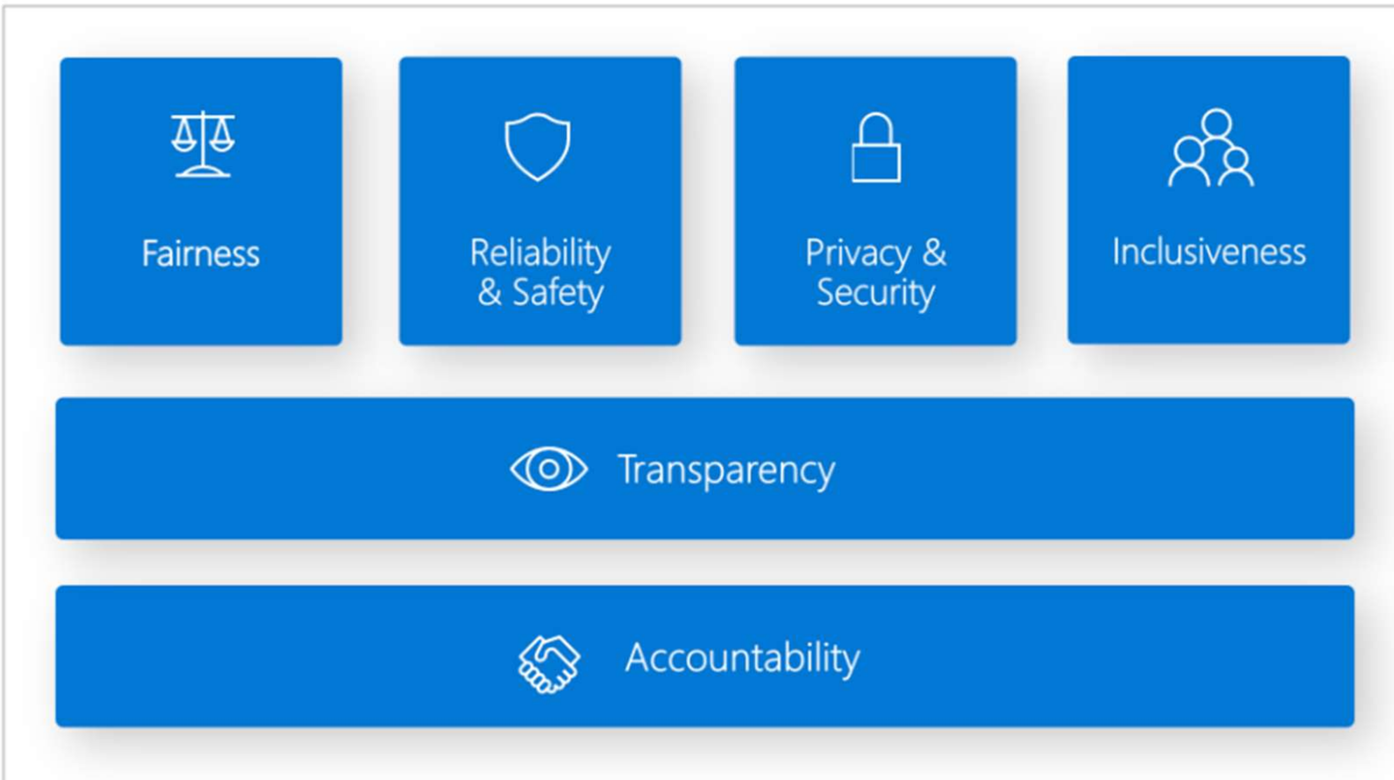


# What is Responsible AI?

Responsible AI, is an innovative methodology that prioritizes the safe, trustworthy, and ethical development, assessment, and deployment of AI systems.

In creating AI systems, developers and deployers make several decisions that significantly impact its outcome.

# What is Responsible AI?



# Why Responsible AI?

By incorporating the principles of Responsible AI :

- These decisions can be guided towards more equitable and beneficial outcomes, thus keeping the system's purpose and human interaction in mind.

# Azure RAI

Microsoft has developed a Responsible AI Standard. It's a framework for building AI systems according to six principles:

- fairness,
- reliability and safety
- privacy and security
- inclusiveness
- transparency
- accountability

# Fairness and inclusiveness

AI systems should treat everyone fairly and avoid affecting similarly situated groups of people in different ways.

For example, when AI systems provide guidance on medical treatment, loan applications, or employment, they should make the same recommendations to everyone who has similar symptoms, financial circumstances, or professional qualifications.

# Fairness and inclusiveness in Azure Machine Learning

The fairness assessment component of the Responsible AI dashboard enables data scientists and developers to assess model fairness across sensitive groups defined in terms of gender, ethnicity, age, and other characteristics.

# Reliability and safety

To build trust, it's critical that AI systems operate reliably, safely, and consistently.

These systems should be able to operate as they were originally designed, respond safely to unanticipated conditions, and resist harmful manipulation.

# Reliability and safety in Azure Machine Learning:

Builds a deep understanding of how failure is distributed for a model.
Identify cohorts (subsets) of data with a higher error rate than the overall benchmark.
These discrepancies might occur when the system or model underperforms for specific demographic groups or for infrequently observed input conditions in the training data.

# Transparency

A crucial part of transparency is interpretability.

it's critical that people understand how those decisions were made by the model.

Improving interpretability requires stakeholders to comprehend how and why AI systems function the way they do.

# Transparency in Azure Machine Learning

The model interpretability and counterfactual what-if components of the Responsible AI dashboard enable data scientists and developers to generate human-understandable descriptions of the predictions of a model.

# Privacy and security

AI systems make accurate and informed predictions and decisions about people using data.

AI systems must require transparency about the collection, use, and storage of data.

Mandate that consumers have appropriate controls to choose how their data is used.

# Privacy and security in Azure Machine Learning

Azure Machine Learning enables administrators and developers to create a secure configuration that complies with their companies' policies.

Restrict access to resources and operations by user account or group.

Restrict incoming and outgoing network communications.

Encrypt data in transit and at rest.Scan for vulnerabilities.
Apply and audit configuration policies.

# Accountability

The people who design and deploy AI systems must be accountable for how their systems operate.

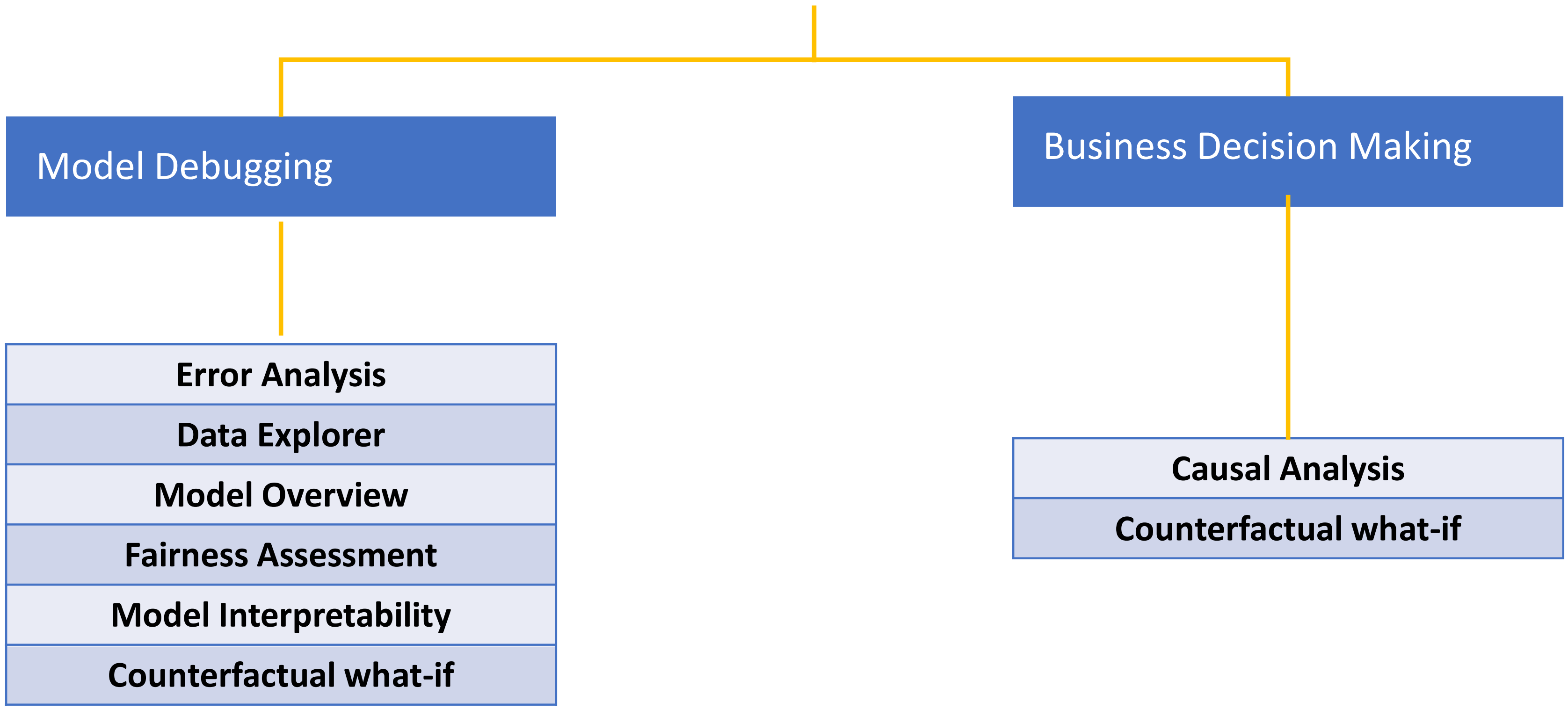
AI systems aren't the final authority on any decision that affects people's lives.

They can also ensure that humans maintain meaningful control over AI systems.

# Accountability in Azure Machine Learning

Machine learning operations (MLOps) is based on DevOps principles.
Register, package, and deploy models and log lineage information.
Track the associated metadata of the model.
Capture the governance data for the end-to-end machine learning lifecycle.
Notify and alert on events in the machine learning lifecycle

# Responsible AI Dashboard




The RAI Insights dashboard constructor and Gather RAI Insights dashboard components are always required,
plus at least one of the tool components. However, it isn't necessary to use all the tools in every Responsible AI dashboard.

# Responsible AI insights in the Azure ML studio

The core components for constructing the Responsible AI dashboard in Azure Machine Learning are:
RAI Insights dashboard constructor

The tool components:

- Add Explanation to RAI Insights dashboard
- Add Causal to RAI Insights dashboard
- Add Counterfactuals to RAI Insights dashboard
- Add Error Analysis to RAI Insights dashboard
- Gather RAI Insights dashboard
- Gather RAI Insights score card

# Model Debugging via Responsible AI dashboard

## Identify

### Error Analysis
Identify cohorts with high error rate versus benchmark and visualize how the error rate distributes

### Fairness Assessment
Evaluate model fairness by exploring a variety of model performance metrics across sensitive groups

## Diagnose

### Model Interpretability
Interpret and debug model.

### Counterfactual Analysis and What If
Generate diverse counterfactual explanations for debugging. Perform feature perturbations

### Exploratory Data Analysis
Understand dataset characteristics

## Mitigate

### Unfairness Mitigation
Mitigate fairness issues (via Fairlearn.org)

### Data Enhancements
Enhance your dataset and retrain model

Model Comparison

## Compare

Backward Compatibility

# Decision Making via Responsible AI dashboard

## Understand data

**Exploratory-Data-Analysis**
Understand dataset characteristics

## Inform Actions

**Causal Inference**
Understand the causal impact of your features on real-world outcomes

**Counterfactual Analysis**
Generate diverse counterfactual explanations for informing end users

# Demo

The next set of slides are representation of Azure RAI Screens which will be demonstrated live in the session using my own Azure Subscription.

We will use the Responsible AI components to assess a regression model & a decision tree model trained on diabetes progression data.
Next, we will walk through the API calls necessary to create a widget with model analysis insights,
We will undertake a visual analysis of the model.

# Plan real-world action using counterfactual example analysis and causal analysis

Launch Responsible AI Toolbox
Train a Model
Create Model and Data Insights
Take Real-World Action
What-If Counterfactuals Analysis
Causal Analysis
Error Analysis

# Snapshots of the code

```
# Enter details of your AML workspace
subscription_id = "<SUBSCRIPTION_ID>"
resource_group = "<RESOURCE_GROUP>"
workspace = "<AML_WORKSPACE_NAME>"
```

```
# Handle to the workspace
from azure.ai.ml import MLClient
from azure.identity import DefaultAzureCredential

credential = DefaultAzureCredential()
ml_client = MLClient(
    credential=credential,
    subscription_id=subscription_id,
    resource_group_name=resource_group,
    workspace_name=workspace,
)
print(ml_client)
```

```
# Get handle to azureml registry for the RAI built in components
registry_name = "azureml"
ml_client_registry = MLClient(
    credential=credential,
    subscription_id=subscription_id,
    resource_group_name=resource_group,
    registry_name=registry_name,
)
print(ml_client_registry)
```

## Accessing the data

First, we need to obtain the dataset and upload it to our AzureML workspace:

```
train_data_path = "data-diabetes-regression/train/"
```

```
test_data_path = "data-diabetes-regression/test/"
```

Load some data for a quick view:

```
import os
import pandas as pd
import mltable

tbl = mltable.load(train_data_path)
train_df: pd.DataFrame = tbl.to_pandas_dataframe()

# test dataset should have less than 5000 rows
test_df = mltable.load(test_data_path).to_pandas_dataframe()
assert len(test_df.index) <= 5000

display(train_df)
```

We are going to create two Datasets in AzureML, one for the train and one for the test datasets.

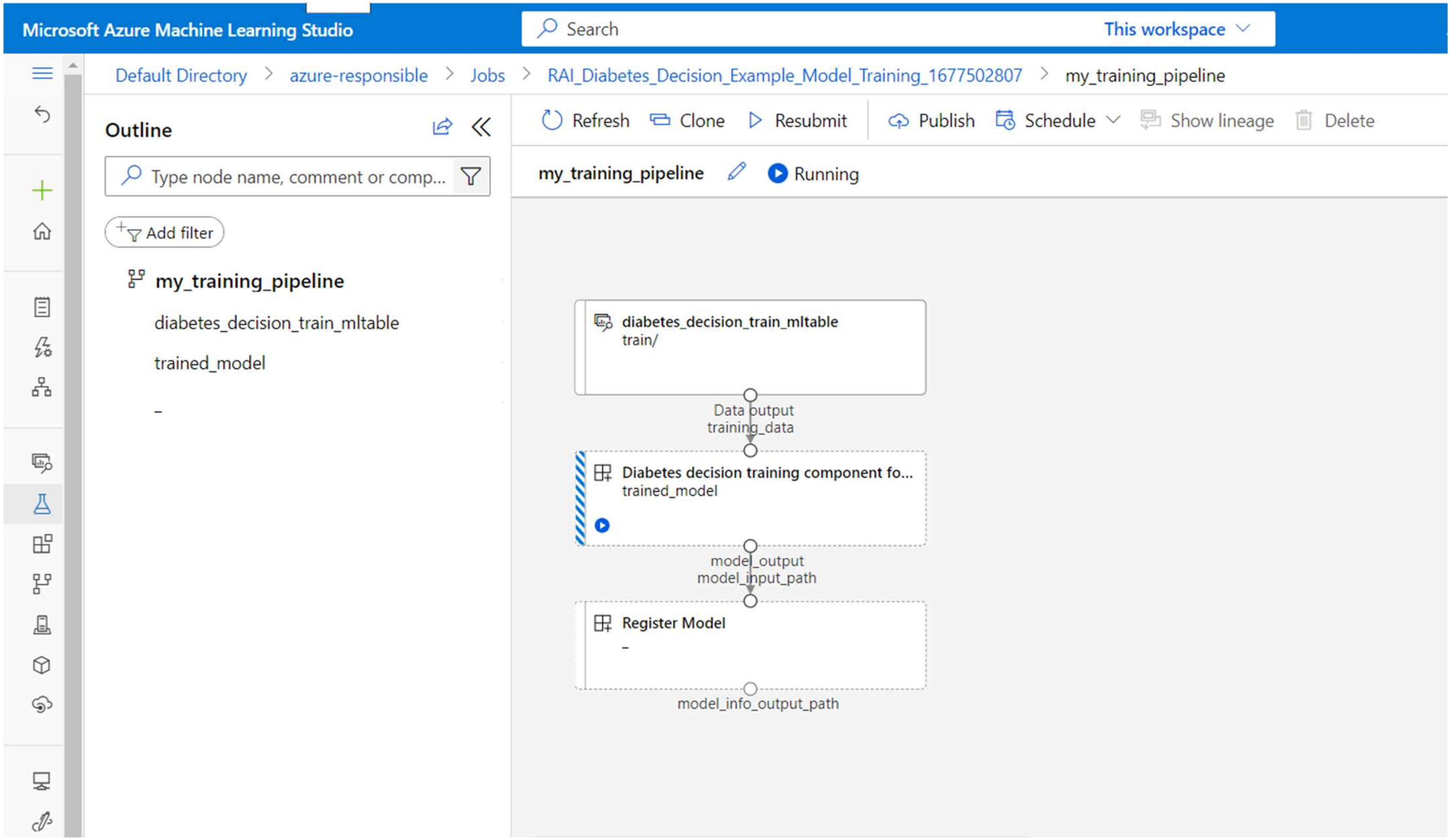
We are going to create two Datasets in AzureML, one for the train and one for the test datasets.

```
from azure.ai.ml.entities import Data
from azure.ai.ml.constants import AssetTypes

input_train_data = "diabetes_regression_train_mltable"
input_test_data = "diabetes_regression_test_mltable"

try:
    # Try getting data already registered in workspace
    train_data = ml_client.data.get(
        name=input_train_data, version=rai_diabetes_regression_example_version_string
    )
    test_data = ml_client.data.get(
        name=input_test_data, version=rai_diabetes_regression_example_version_string
    )
except Exception as e:
    train_data = Data(
        path=train_data_path,
        type=AssetTypes.MLTABLE,
        description="RAI diabetes regression example training data",
        name=input_train_data,
        version=rai_diabetes_regression_example_version_string,
    )
    ml_client.data.create_or_update(train_data)

    test_data = Data(
        path=test_data_path,
        type=AssetTypes.MLTABLE,
        description="RAI diabetes regression example test data",
        name=input_test_data,
        version=rai_diabetes_regression_example_version_string,
    )
    ml_client.data.create_or_update(test_data)
```

# Model training

A model training pipeline : We will use a Azure ML Studio pipeline. This will have two stages:
The actual training component
A model registration component

The training component :We will train a RandomForestRegressor on the input data and save it using MLFlow and scikit-learn ML libraries.


Microsoft Azure Machine Learning Studio
Search
This workspace
Default Directory > azure-responsible > Jobs > RAI_Diabetes_Decision_Example_Model_Training_1677502807 > my_training_pipeline
Outline
Type node name, comment or comp...
Add filter
my_training_pipeline
diabetes_decision_train_mltable
trained_model
–
Refresh
Clone
Resubmit
Publish
Schedule
Show lineage
Delete
my_training_pipeline
Running
diabetes_decision_train_mltable
train/
Data output
training_data
Diabetes decision training component fo...
trained_model
model_output
model_input_path
Register Model
–
model_info_output_path

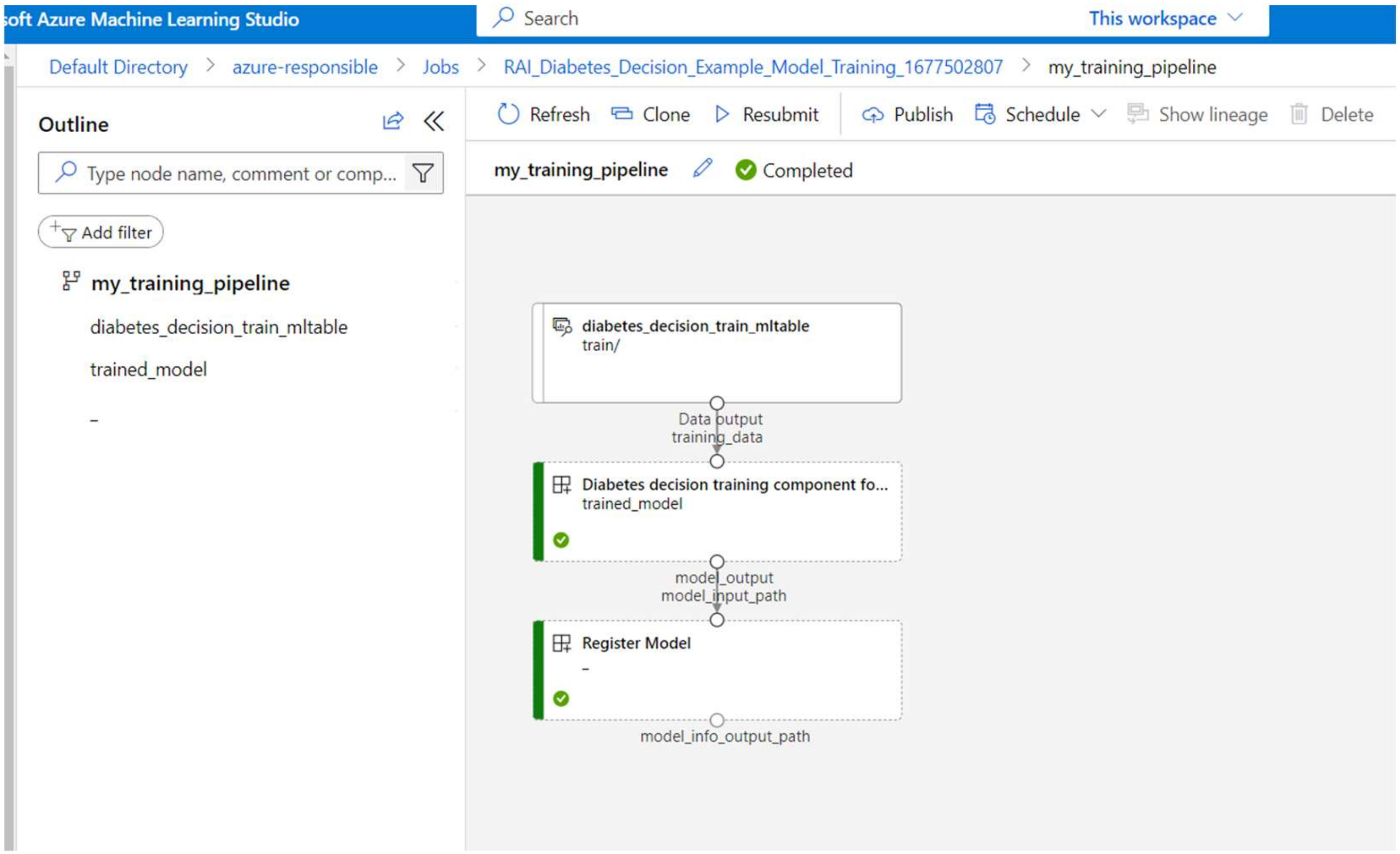
```python
from azure.ai.ml.entities import PipelineJob
from IPython.core.display import HTML
from IPython.display import display


def submit_and_wait(ml_client, pipeline_job) -> PipelineJob:
    created_job = ml_client.jobs.create_or_update(pipeline_job)
    assert created_job is not None

    print("Pipeline job can be accessed in the following URL:")
    display(HTML('<a href="{0}">{0}</a>'.format(created_job.studio_url)))

    while created_job.status not in [
        "Completed",
        "Failed",
        "Canceled",
        "NotResponding",
    ]:
        time.sleep(30)
        created_job = ml_client.jobs.get(created_job.name)
        print("Latest status : {0}".format(created_job.status))
    assert created_job.status == "Completed"
    return created_job


# This is the actual submission
training_job = submit_and_wait(ml_client, model_registration_pipeline_job)
```

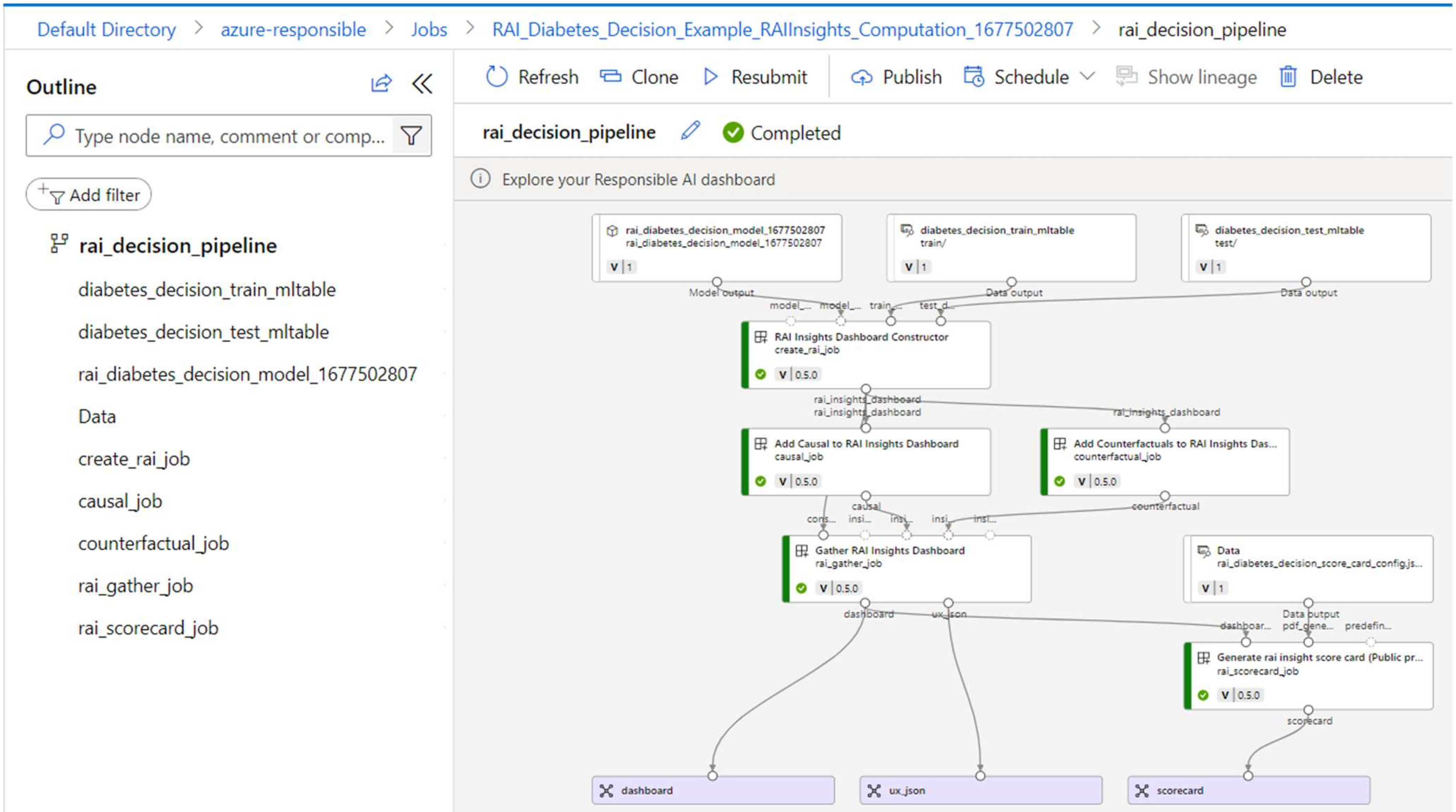
```
Uploading component_src (0.0 MBs): 100%|██████████| 2437/2437 [00:00<00:00, 271084.68it/s]

Uploading register_model_src (0.0 MBs): 100%|██████████| 2394/2394 [00:00<00:00, 240720.25it/s]

Pipeline job can be accessed in the following URL:
```

https://ml.azure.com/runs/strong_grape_nlkf9hk9km?wsid=/subscriptions/3af4d623-0c76-44c2-b92c-7593399a5c
rg/workspaces/azure-responsible&tid=bfd1f0c0-cd46-4cc3-84fd-8a6c2d01904d

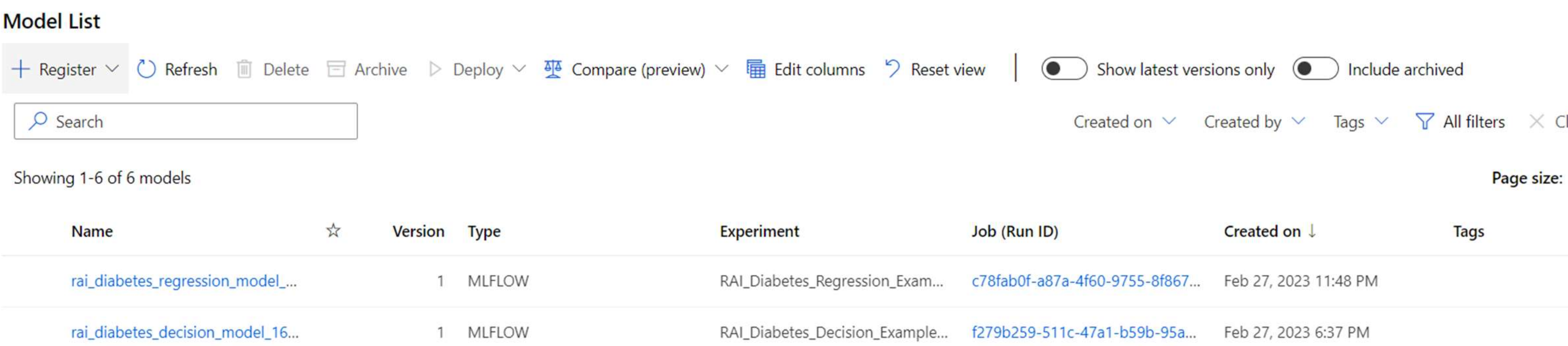
```
Latest status : Running
Latest status : Running
Latest status : Running
Latest status : Running
Latest status : Running
Latest status : Running
Latest status : Running
Latest status : Running
Latest status : Running
Latest status : Running
Latest status : Running
Latest status : Running
Latest status : Running
Latest status : Running
Latest status : Running
Latest status : Completed
```

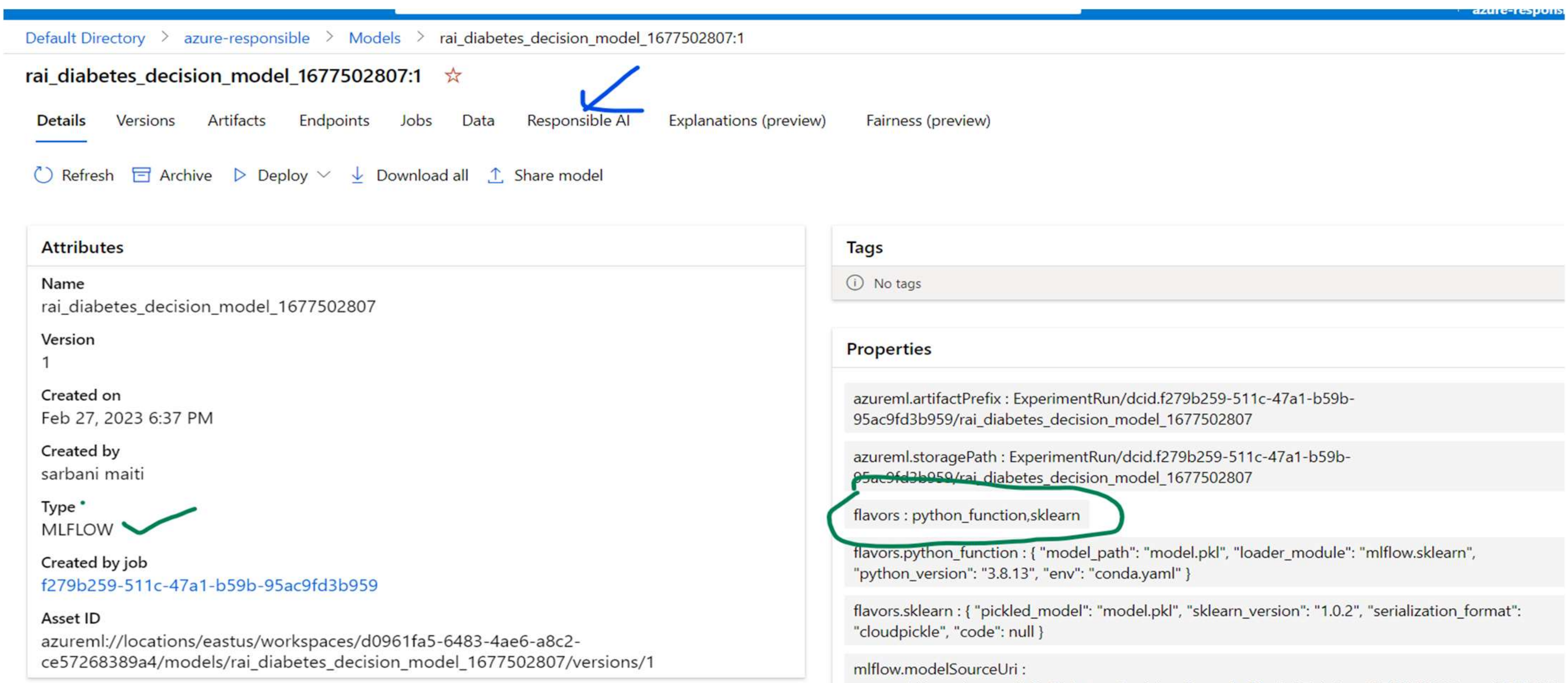
# Model Pipelines

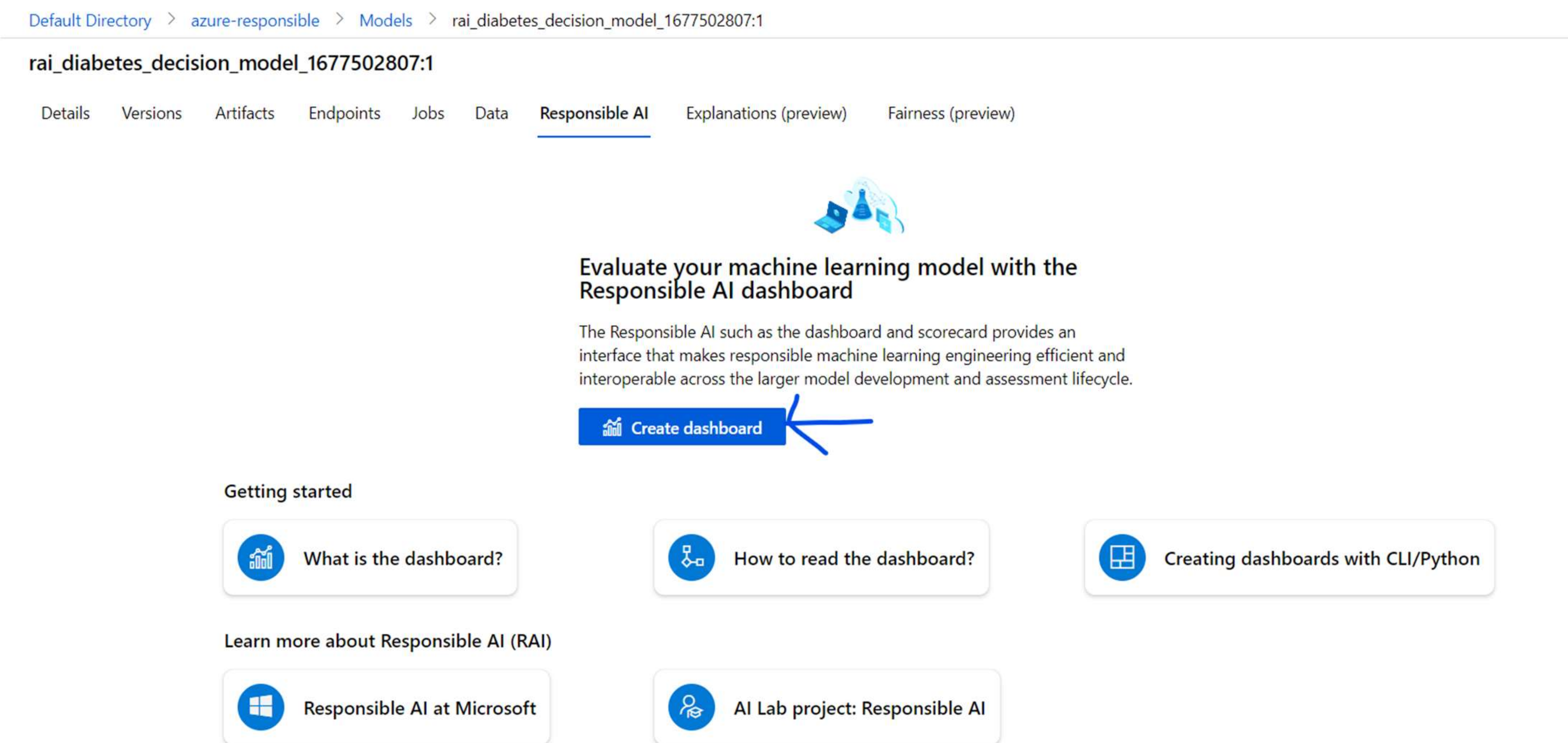



Default Directory > azure-responsible > Jobs > RAI_Diabetes_Decision_Example_RAIInsights_Computation_1677502807 > rai_decision_pipeline
Refresh
Clone
Resubmit
Publish
Schedule
Show lineage
Delete
Outline
Type node name, comment or comp...
Add filter
rai_decision_pipeline
diabetes_decision_train_mltable
diabetes_decision_test_mltable
rai_diabetes_decision_model_1677502807
Data
create_rai_job
causal_job
counterfactual_job
rai_gather_job
rai_scorecard_job
rai_decision_pipeline
Completed
Explore your Responsible AI dashboard
rai_diabetes_decision_model_1677502807
diabetes_decision_train_mltable
train/
diabetes_decision_test_mltable
test/
Model output
Data output
RAI Insights Dashboard Constructor
create_rai_job
rai_insights_dashboard
Add Causal to RAI Insights Dashboard
causal_job
Add Counterfactuals to RAI Insights Das...
counterfactual_job
causal
counterfactual
Gather RAI Insights Dashboard
rai_gather_job
Data
rai_diabetes_decision_score_card_config.js...
dashboard
ux_json
Generate rai insight score card (Public pr...
rai_scorecard_job
scorecard
dashboard
ux_json
scorecard

# Azure ML Studio – Registered Model




Default Directory > azure-responsible > Models > rai_diabetes_decision_model_1677502807:1
rai_diabetes_decision_model_1677502807:1
Details
Versions
Artifacts
Endpoints
Jobs
Data
Responsible AI
Explanations (preview)
Fairness (preview)
Refresh
Archive
Deploy
Download all
Share model
Attributes
Name
rai_diabetes_decision_model_1677502807
Version
1
Created on
Feb 27, 2023 6:37 PM
Created by
sarbani maiti
Type
MLFLOW
Created by job
f279b259-511c-47a1-b59b-95ac9fd3b959
Asset ID
azureml://locations/eastus/workspaces/d0961fa5-6483-4ae6-a8c2-ce57268389a4/models/rai_diabetes_decision_model_1677502807/versions/1
Tags
No tags
Properties
azureml.artifactPrefix : ExperimentRun/dcid.f279b259-511c-47a1-b59b-95ac9fd3b959/rai_diabetes_decision_model_1677502807
azureml.storagePath : ExperimentRun/dcid.f279b259-511c-47a1-b59b-95ac9fd3b959/rai_diabetes_decision_model_1677502807
flavors : python_function,sklearn
flavors.python_function : { "model_path": "model.pkl", "loader_module": "mlflow.sklearn", "python_version": "3.8.13", "env": "conda.yaml" }
flavors.sklearn : { "pickled_model": "model.pkl", "sklearn_version": "1.0.2", "serialization_format": "cloudpickle", "code": null }
mlflow.modelSourceUri :


Default Directory > azure-responsible > Models > rai_diabetes_decision_model_1677502807:1
rai_diabetes_decision_model_1677502807:1
Details
Versions
Artifacts
Endpoints
Jobs
Data
Responsible AI
Explanations (preview)
Fairness (preview)
Evaluate your machine learning model with the Responsible AI dashboard
The Responsible AI such as the dashboard and scorecard provides an interface that makes responsible machine learning engineering efficient and interoperable across the larger model development and assessment lifecycle.
Create dashboard
Getting started
What is the dashboard?
How to read the dashboard?
Creating dashboards with CLI/Python
Learn more about Responsible AI (RAI)
Responsible AI at Microsoft
AI Lab project: Responsible AI

# Code sample

And submit the pipeline to AzureML for execution:

```
insights_job = submit_and_wait(ml_client, insights_pipeline_job)
```

The dashboard should appear in the AzureML portal in the registered model view. The following cell computes the expected URI:

```
sub_id = ml_client._operation_scope.subscription_id
rg_name = ml_client._operation_scope.resource_group_name
ws_name = ml_client.workspace_name

expected_uri = f"https://ml.azure.com/model/{expected_model_id}/model_analysis?wsid=/subscriptions/{sub_id}/resourcegroups/{rg_name}/workspaces/{ws_

print(f"Please visit {expected_uri} to see your analysis")
```

```
from azure.ai.ml import load_component

yaml_contents = (
    f"""
$schema: http://azureml/sdk-2-0/CommandComponent.json
name: rai_diabetes_regression_training_component
display_name: Diabetes regression training component for RAI example
version: {rai_diabetes_regression_example_version_string}
type: command
inputs:
  training_data:
    type: path
  target_column_name:
    type: string
outputs:
  model_output:
    type: path
code: ./component_src/
environment: azureml://registries/azureml/environments/AzureML-responsibleai-0.20-ubuntu20.04-py38-cpu/versions/4
"""
    + r"""
command: >-
  python diabetes_regression_training_script.py
  --training_data ${{{{inputs.training_data}}}}
  --target_column_name ${{{{inputs.target_column_name}}}}
  --model_output ${{{{outputs.model_output}}}}
"""
)

yaml_filename = "RAIDiabetesRegressionTrainingComponent.yaml"

with open(yaml_filename, "w") as f:
    f.write(yaml_contents.format(yaml_contents))

train_model_component = load_component(source=yaml_filename)
```

```
import uuid
from azure.ai.ml import Output

# Pipeline to construct the RAI Insights
insights_pipeline_job = rai_regression_pipeline(
    target_column_name=target_feature,
    train_data=diabetes_train_pq,
    test_data=diabetes_test_pq,
    score_card_config_path=score_card_config_path,
)

# Workaround to enable the download
rand_path = str(uuid.uuid4())
insights_pipeline_job.outputs.dashboard = Output(
    path=f"azureml://datastores/workspaceblobstore/paths/{rand_path}/dashboard/",
    mode="upload",
    type="uri_folder",
)
insights_pipeline_job.outputs.ux_json = Output(
    path=f"azureml://datastores/workspaceblobstore/paths/{rand_path}/ux_json/",
    mode="upload",
    type="uri_folder",
)
insights_pipeline_job.outputs.scorecard = Output(
    path=f"azureml://datastores/workspaceblobstore/paths/{rand_path}/scorecard/",
    mode="upload",
    type="uri_folder",
)
```

# RAI Dashboard with Causal Insight + Counterfactual

Default Directory > azure-responsible > Models > rai_diabetes_decision_model_1677502807:1

**rai_diabetes_decision_model_1677502807:1**

Details Versions Artifacts Endpoints Jobs Data **Responsible AI** Explanations (preview) Fairness (preview)

The Responsible AI dashboard provides a single dashboard to help you debug and assess your machine learning models responsible, while informing your data-driven decisions. Learn more. Responsible AI insights can be generated from the SDK/CLI or through the UI wizard and viewed with your Responsible AI dashboard. To view your insights, click on the name of your dashboard. Different combinations of components (explainers, causal analysis, etc.) can be attached to each Responsible AI dashboard. The list below only shows whether or not a component was generated for your dashboard, but different components can be viewed or hidden within the dashboard itself. Learn more.

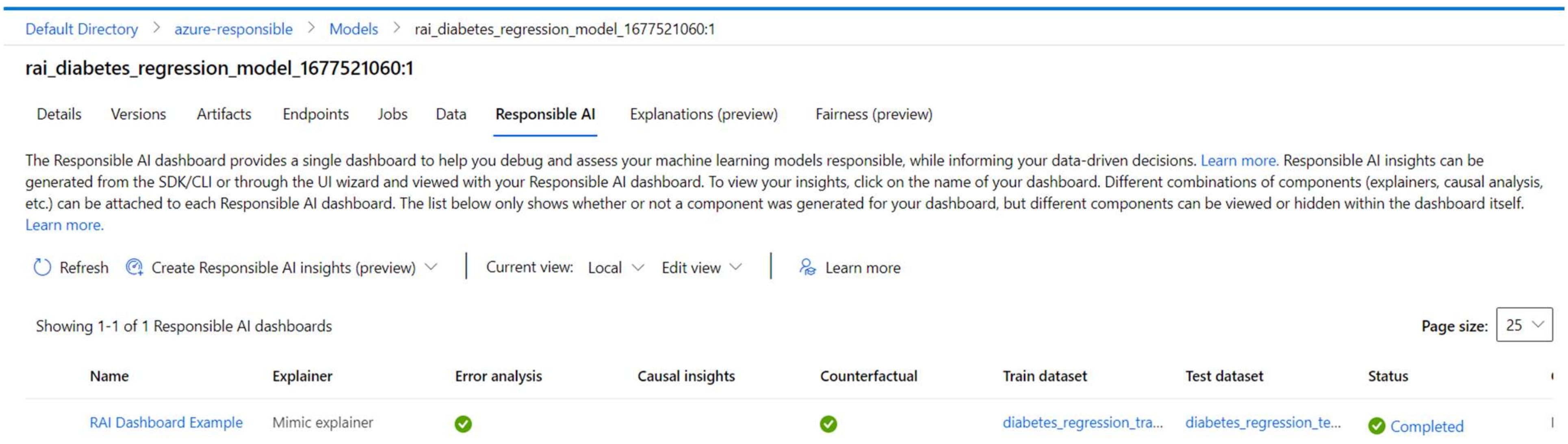
Refresh Create Responsible AI insights (preview) | Current view: Local Edit view | Learn more

Showing 1-1 of 1 Responsible AI dashboards

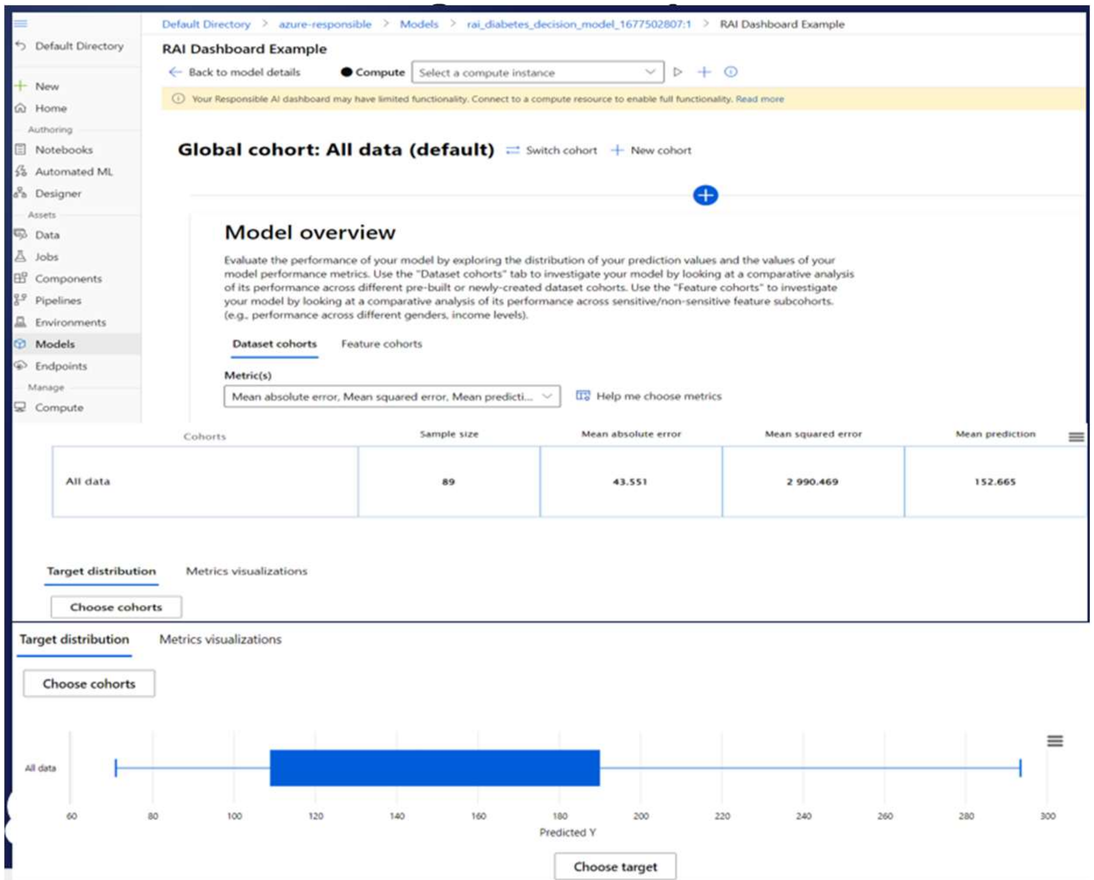
**Page size:** 25

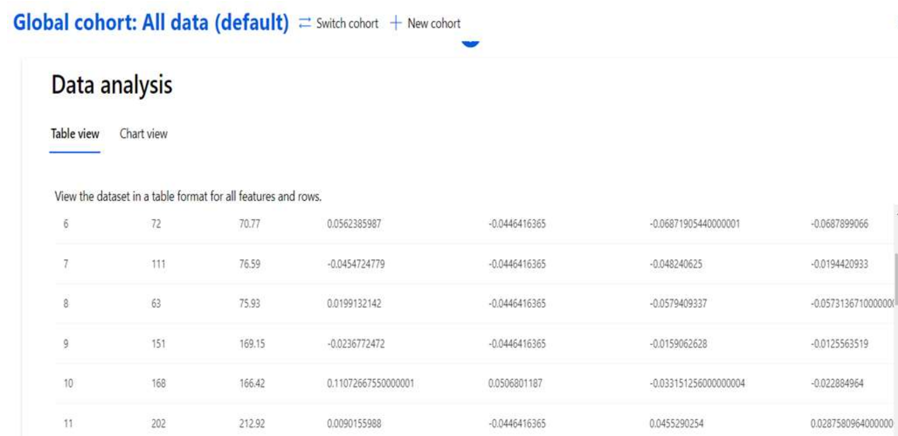
| Name | Explainer | Error analysis | Causal insights | Counterfactual | Train dataset | Test dataset | Status |
|---|---|---|---|---|---|---|---|
| RAI Dashboard Example | | | ✔ | ✔ | diabetes_decision_train... | diabetes_decision_test_... | Completed |

# RAI Dashboard with Error Analysis+ Counterfactual



# RAI Dashboard with Causal Insight +



# How to take decision : What-If Counterfactuals Analysis

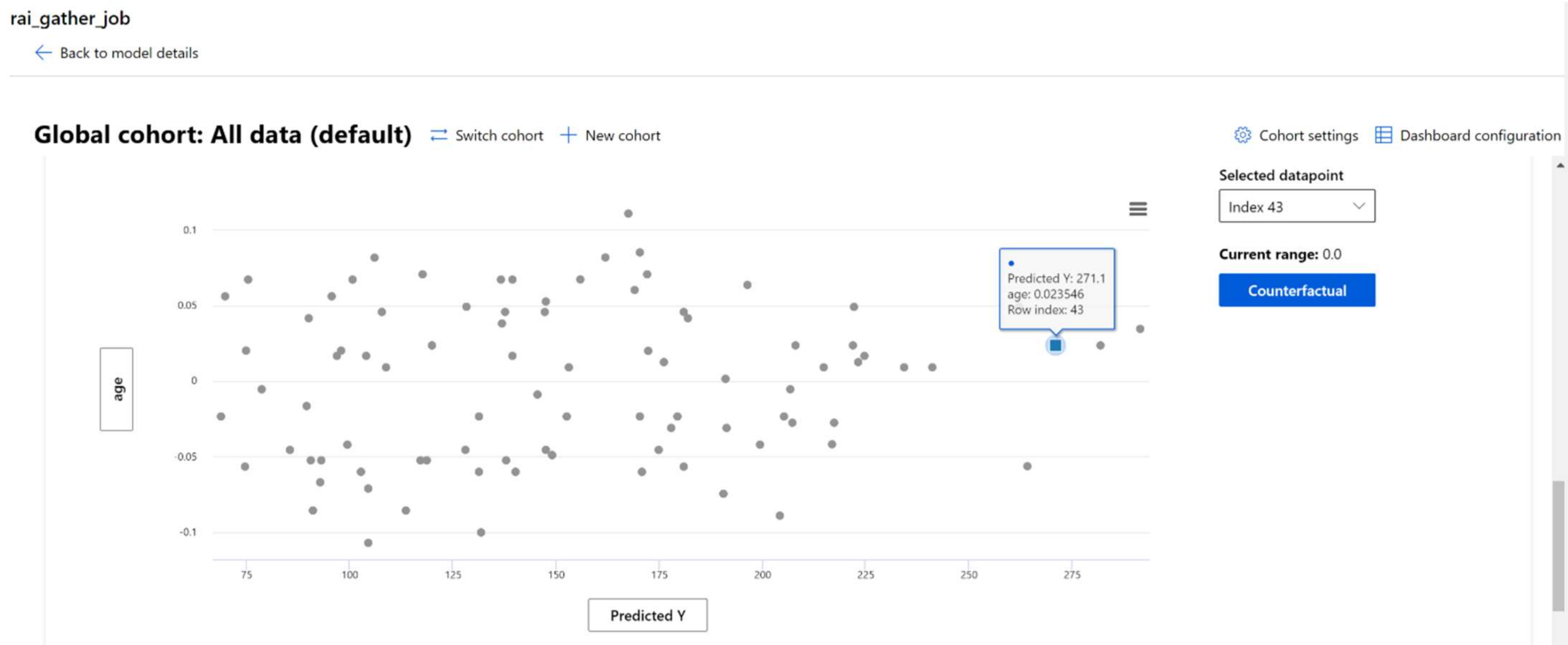
Let's imagine that the diabetes progression scores predicted by the model are used to determine medical insurance rates .

If the score is greater than 120, there is a higher rate.

Let’s take 43rd Patient sample who had scored 271.1 in this increased rate, and they want to know how they should change their health to get a lower rate prediction from the model.

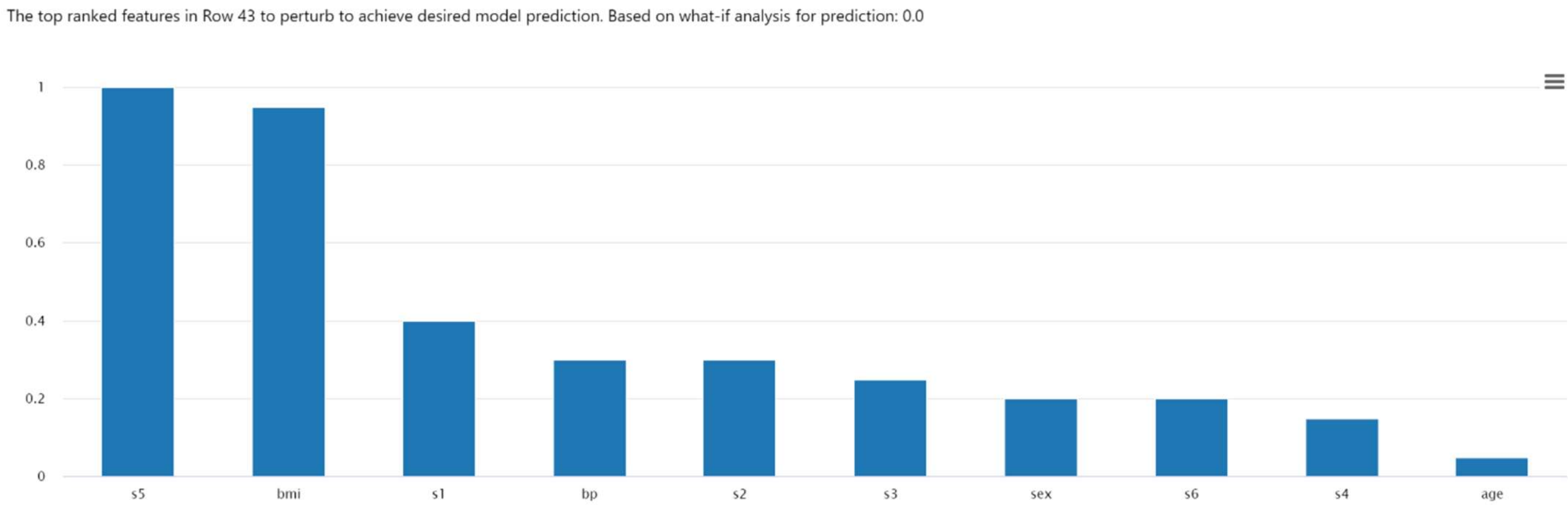
Also the patient wants to get lower insurance price.

# Counterfactuals Analysis with RAI Dashboard




The What-If counterfactuals component shows how slightly different feature values affect model predictions. This can be used to solve Patient 43's problem.

# Counterfactuals Analysis with RAI Dashboard




The top ranked features bar plot shows that bmi and s5 are the best two features to bring the model score within 120 for the patient .

# Causal Analysis

Now suppose that a doctor wishes to know how to reduce the progression of diabetes in her patients. This can be explored in the Causal Inference component of the Responsible AI Toolbox.

In the "Aggregate causal effects" tab, it is possible to see how perturbing features causes lower disease progression.

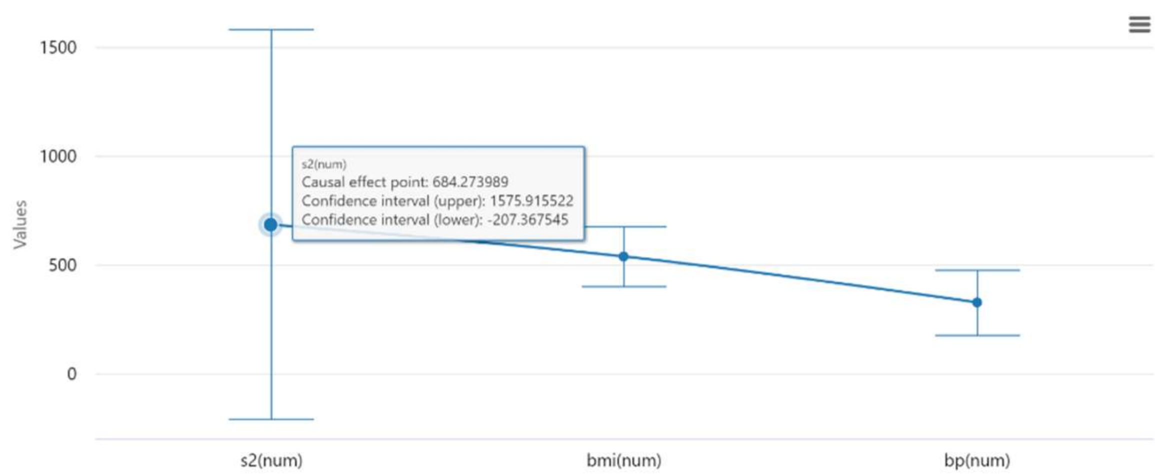
# Causal Analysis – RAI Dashboard

rai_gather_job

← Back to model details

**Global cohort: All data (default)** Switch cohort + New cohort Cohort settings Dashboard configuration

| Feature | Effect estimate | Standard error | Z-score | P-value | Confidence interv... | Confidence interval (upper) |
|---|---|---|---|---|---|---|
| s2(num) | 6.843e+2 | 4.549e+2 | 1.504e+0 | 1.325e-1 | -2.074e+2 | 1.576e+3 |
| bmi(num) | 5.386e+2 | 7.103e+1 | 7.583e+0 | 3.364e-14 | 3.994e+2 | 6.778e+2 |
| bp(num) | 3.281e+2 | 7.633e+1 | 4.298e+0 | 1.723e-5 | 1.785e+2 | 4.777e+2 |




**Continuous treatments:** On average in this sample, increasing this feature by 1 unit will cause the probability of class/label 1 to increase by X units.

**Binary treatments:** On average in this sample, turning on this feature will cause the probability of class/label 1 to increase by X units.

A lasso (or logistic regression if y is binary) was fit to predict y from X[-i], and a lasso (or logistic regression if X[i] is categorical) was fit to predict X[i] from X[-i]. The causal effect can be viewed as the average correlation of the residuals/surprise variation of the two prediction tasks. Learn more about Double Machine Learning here

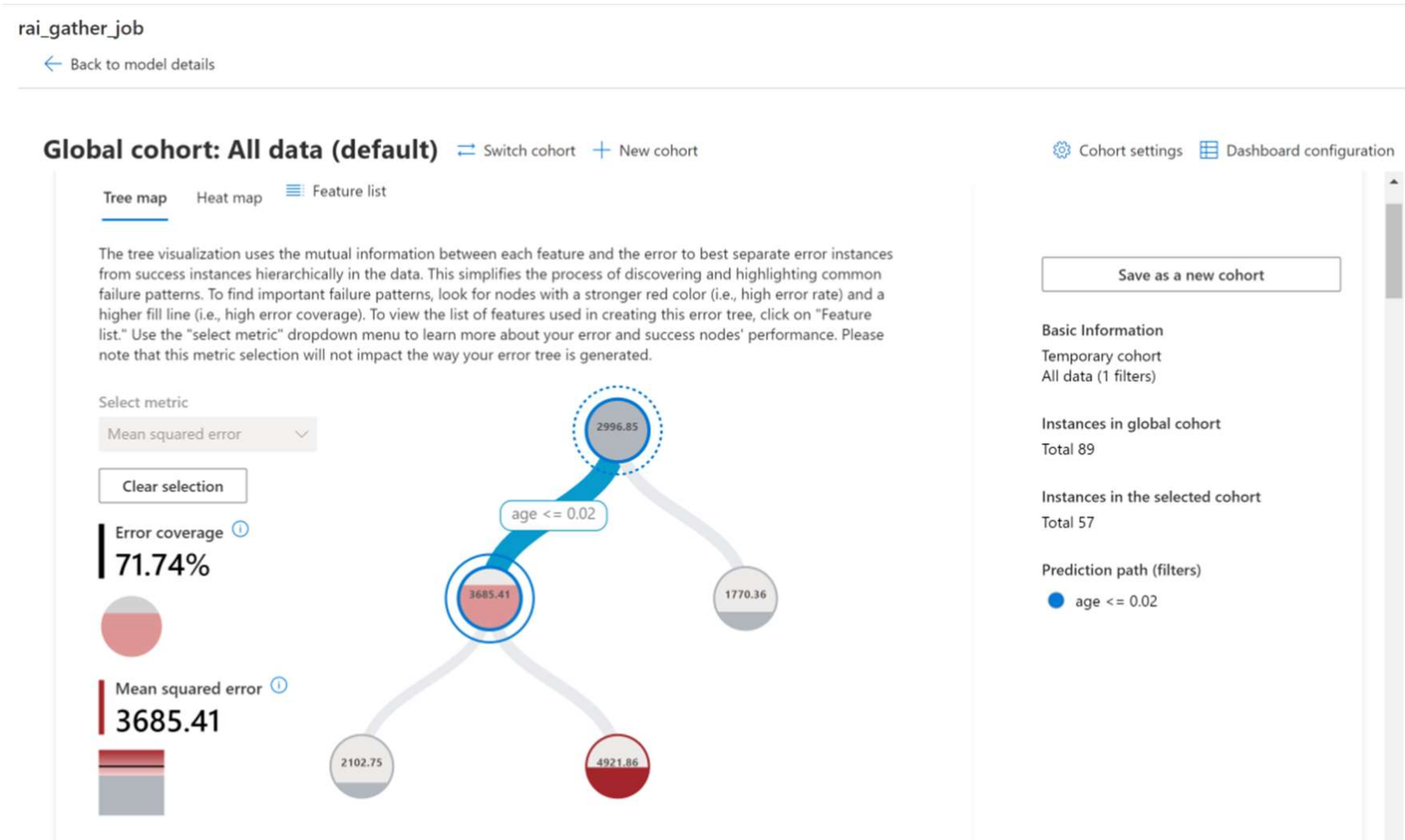
It appears that increasing s2 (LDL) by one unit, would increase diabetes progression by around 684 units (again, exact values can vary due to random number effects):

# Error Analysis : Aggregate Analysis

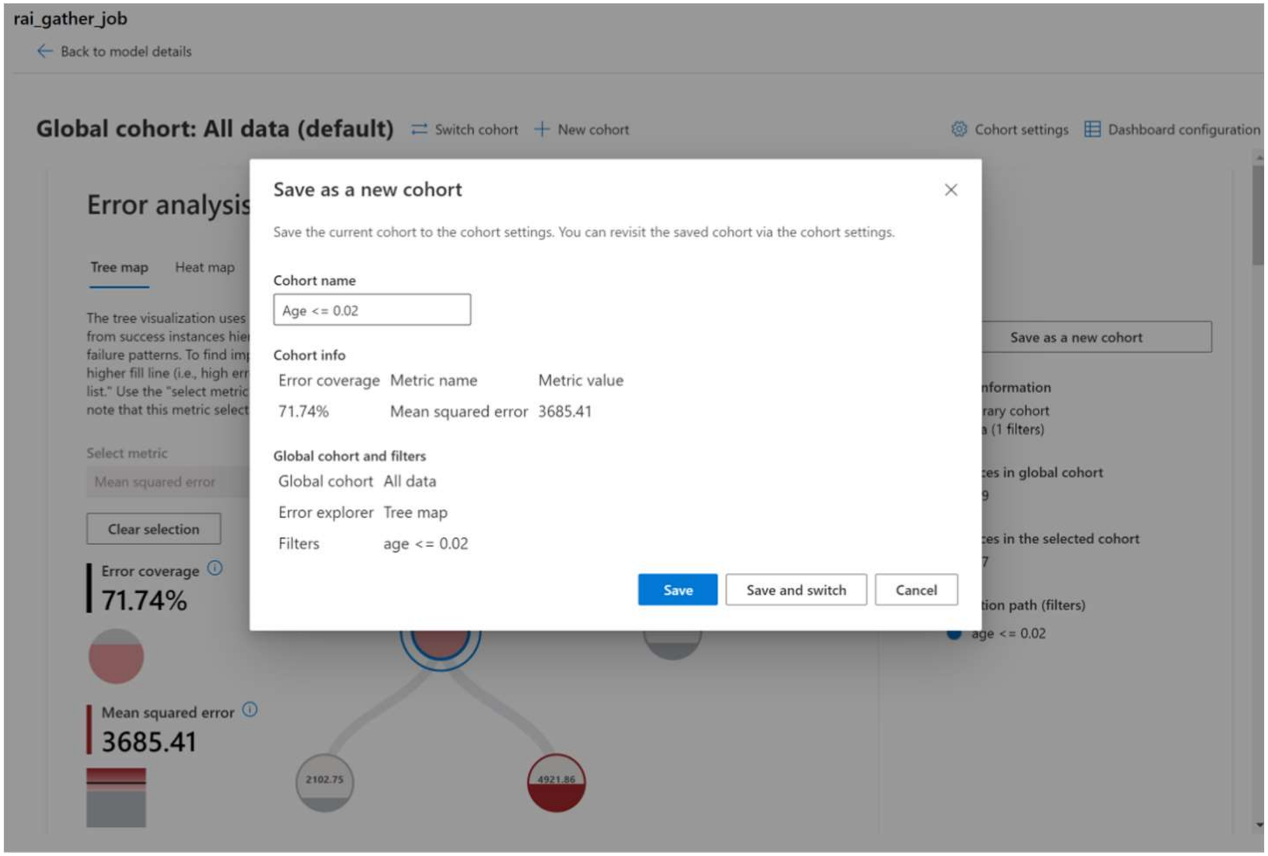
Upon opening the RAI dashboard in the AzureML portal, the Error Analysis component is displayed at the top.

The tree map view of this component visualizes the cohort breakdown of error in nodes:

# Error Analysis : RAI Dashboard



# Error Analysis : RAI Dashboard




For this model, over 70% of the error is concentrated in datapoints whose age feature is less than 0.02. Note that this value has been mean-centered and scaled by the number of samples * standard deviation.

We can explore this cohort further by saving the cohort of interest.

# Model Explainability

Explain the entire model behavior or individual predictions on your personal machine locally.

Enable interpretability techniques for engineered features.

Explain the behavior for the entire model and individual predictions in Azure.

Upload explanations to Azure Machine Learning Run History.

Use a visualization dashboard to interact with your model explanations, both in a Jupyter Notebook and in the Azure Machine Learning studio.

Deploy a scoring explainer alongside your model to observe explanations during inferencing.

# Model Explainability : Python SDK

1. Install the `azureml-interpret` package.

Bash    Copy

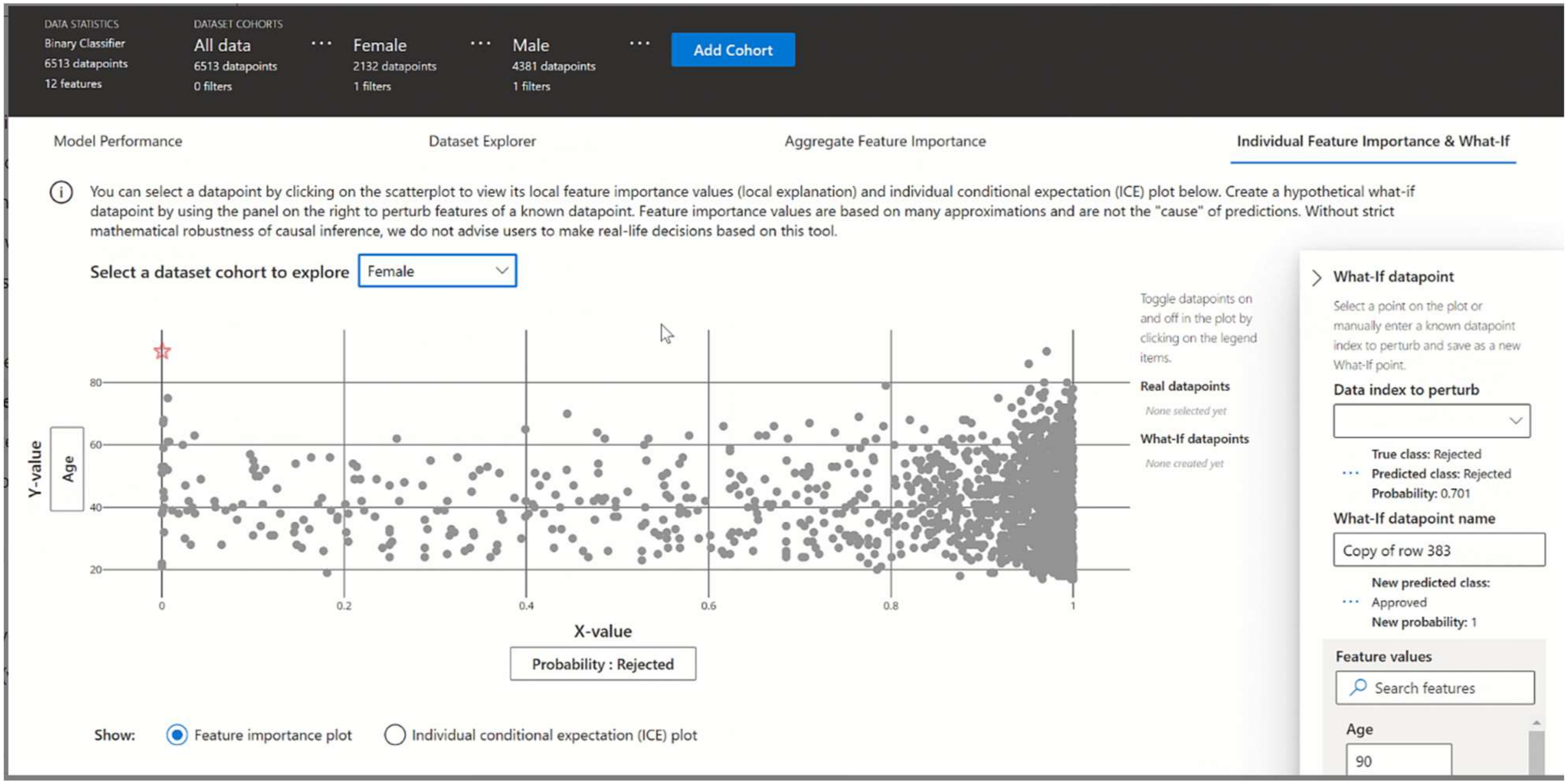
```bash
pip install azureml-interpret
```

2. Train a sample model in a local Jupyter Notebook.

Python    Copy

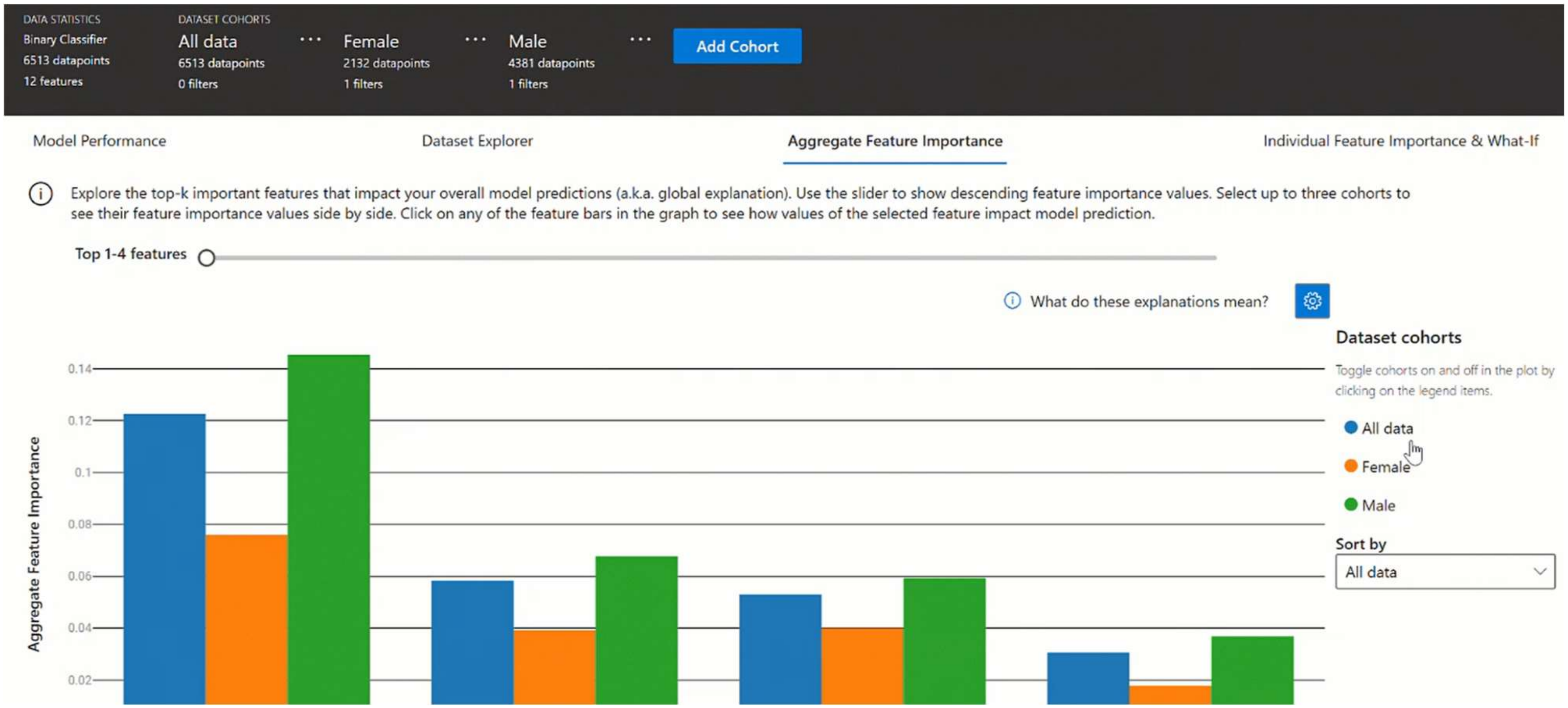
```python
# load breast cancer dataset, a well-known small dataset that comes with scikit-learn
from sklearn.datasets import load_breast_cancer
from sklearn import svm
from sklearn.model_selection import train_test_split
breast_cancer_data = load_breast_cancer()
classes = breast_cancer_data.target_names.tolist()

# split data into train and test
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(breast_cancer_data.data,
                                                    breast_cancer_data.target,
                                                    test_size=0.2,
                                                    random_state=0)
clf = svm.SVC(gamma=0.001, C=100., probability=True)
model = clf.fit(x_train, y_train)
```

# Visualizations through RAI Dashboard

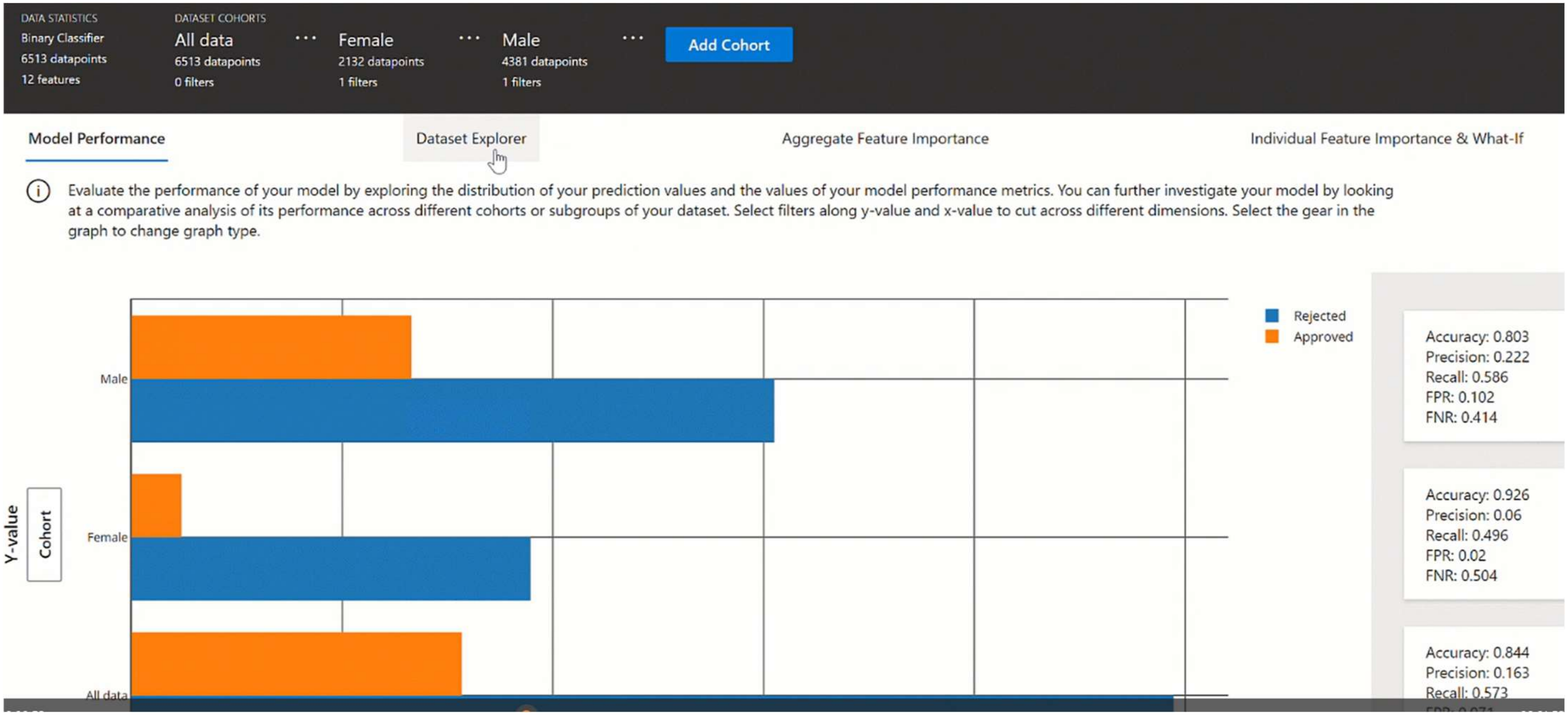


# Visualizations through RAI Dashboard

# Visualizations through RAI Dashboard



# Visualizations through RAI Dashboard



# References

Download the score card reports.

Code references:

https://github.com/sarbaniAi/WiDSWorkshop_28Jun2023

https://github.com/Azure/azureml-examples/blob/main/sdk/python/responsible-ai/responsibleaidashboard-diabetes-decision-making/responsibleaidashboard-diabetes-decision-making.ipynb

https://github.com/Azure/azureml-examples/tree/main/sdk/python/responsible-ai/responsibleaidashboard-diabetes-regression-model-debugging

Thank you