



CDW Documentation

Responsible AI Test

Responsible AI Test

Purpose

Evaluate the Responsible AI dashboard and see what it does.

Test Process

Here's a structured list of **Responsible AI Dashboard Deployment Steps** using the corrected scripts. Each step includes:

- **Step Number & Action**
- **Purpose**
- **Expected Result**

Step 1: Install Required Packages

```
pip install --upgrade raiutils raiwidgets responsibleai ipywidgets
```

Purpose:

Install the Python packages required to run Responsible AI analysis and render the dashboard.

Expected Result:

Packages are installed without errors; dashboard widgets can render in the notebook (after kernel restart).

Step 2: Load and Preprocess the Dataset

```
from sklearn.datasets import fetch_openml
import pandas as pd

data = fetch_openml(name='adult', version=2, as_frame=True)
df = data.frame.dropna()
```

Purpose:

Load a well-known classification dataset (Adult Census Income) and remove any missing values to avoid downstream errors.

Expected Result:

A clean DataFrame with no null values is loaded.

□ Step 3: Split Dataset into Train and Test Sets

```
from sklearn.model_selection import train_test_split

target_column = 'class'
X = df.drop(columns=[target_column])
y = df[target_column]

X_train, X_test, y_train, y_test = train_test_split(X, y, stratify=y,
random_state=42)
```

□ Purpose:

Separate features and target, then split into training/testing sets for model training and evaluation.

□ Expected Result:

X_train, X_test, y_train, y_test variables created and stratified properly.

□ Step 4: Define Preprocessing and Train a Model

```
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import OneHotEncoder, StandardScaler
from sklearn.compose import ColumnTransformer
from sklearn.ensemble import RandomForestClassifier

categorical_cols = X_train.select_dtypes(include=['object',
'category']).columns.tolist()
numerical_cols = X_train.select_dtypes(include=['int64',
'float64']).columns.tolist()

preprocessor = ColumnTransformer([
    ('cat', OneHotEncoder(handle_unknown='ignore'), categorical_cols),
    ('num', StandardScaler(), numerical_cols)
])

clf = Pipeline(steps=[
    ('preprocessor', preprocessor),
    ('classifier', RandomForestClassifier(n_estimators=10, random_state=42))
])

clf.fit(X_train, y_train)
```

□ Purpose:

Build a model pipeline that encodes categorical features, scales numeric ones, and trains a classifier.

□ Expected Result:

Pipeline is trained successfully on the training data without conversion errors.

□ Step 5: Prepare Data for RAIInsights

```
# Ensure target column is a supported type
y_train_clean = y_train.astype(str)
y_test_clean = y_test.astype(str)

train_data = X_train.copy()
train_data[target_column] = y_train_clean

test_data = X_test.copy()
test_data[target_column] = y_test_clean
```

□ Purpose:

Re-attach the target column (as string) to the feature DataFrames — required for RAIInsights.

□ Expected Result:

train_data and test_data DataFrames contain all required columns including the target.

□ Step 6: Initialize the Responsible AI Insights Object

```
from responsibleai import RAIInsights, FeatureMetadata

feature_metadata = FeatureMetadata(categorical_features=categorical_cols)

rai_insights = RAIInsights(
    model=clf,
    train=train_data,
    test=test_data,
    target_column=target_column,
    task_type="classification",
    feature_metadata=feature_metadata
)
```

□ Purpose:

Create a RAIInsights object that acts as the core engine for the Responsible AI dashboard.

□ Expected Result:

RAIInsights object is initialized successfully and ready for configuration.

□ Step 7: Add Responsible AI Analysis Tools

```
rai_insights.explainer.add()
rai_insights.error_analysis.add()
```

```
rai_insights.counterfactual.add(total_CFs=10, desired_class='opposite')
rai_insights.causal.add(treatment_features=categorical_cols)
```

□ **Purpose:**

Attach various tools (explanation, error analysis, counterfactuals, causal inference) to the insights engine.

□ **Expected Result:**

No errors thrown; tools are queued for computation.

□ **Step 8: Compute Insights**

```
rai_insights.compute()
```

□ **Purpose:**

Run analysis for all selected tools. This step may take a minute or more.

□ **Expected Result:**

Tool outputs are generated for the first 5,000 rows of the test set.

□ **Step 9: Launch the Responsible AI Dashboard**

```
from raiwidgets import ResponsibleAIDashboard

ResponsibleAIDashboard(rai_insights)
```

□ **Purpose:**

Open an interactive dashboard to explore insights such as feature importance, what-if analysis, and error breakdowns.

□ **Expected Result:**

A dashboard is displayed inside the notebook. Interactive plots and controls are available for analysis.

NOTE: Due to the way that these URLs are deployed, this step will fail because the notebook sends the wrong headers and this is expected. You have to either pull the notebook local and use it from the terminal or register the dashboard/dataset and review it through the portal.

NOTE: To deploy locally you need to follow the process below.

Use Local Jupyter Notebook

1. Download the full notebook (.ipynb) to your local machine.
2. Create a conda/venv environment with:

```
pipx install raiwidgets responsibleai scikit-learn ipywidgets jupyter
```

notebooks

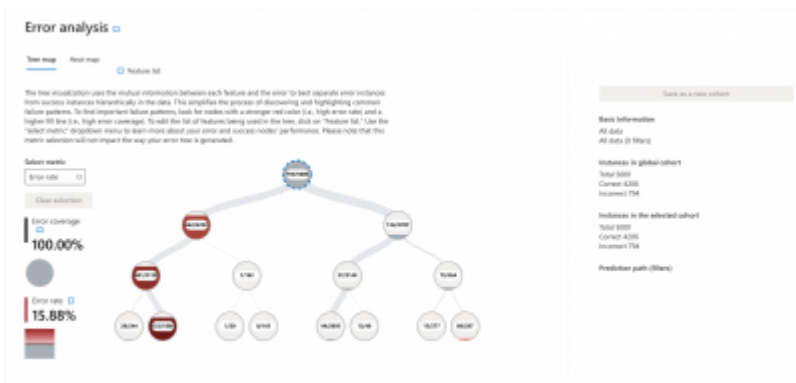
3. Launch local notebook from the directory you downloaded the file to using:

jupyter notebook

1. This will launch a notebook session in your default browser.
2. Open downloaded ipynb file.
3. Rerun all steps in local notebook.

☐ It will render **inline** without CORS issues.

Output



The screenshot shows the 'Data analysis' interface in 'Table view'. It displays a table with columns: Index, True?, Predicted?, age, workclass, bkgct, education, education-num, and sex. The table lists 12 rows of data, including fields like 'age', 'workclass', 'bkgct', 'education', and 'education-num'.

This is another screenshot of the 'Data analysis' interface in 'Table view', showing the same table structure as the previous one. It displays 12 rows of data with columns for Index, True?, Predicted?, age, workclass, bkgct, education, education-num, and sex.

Feature importances

Aggregate feature importance Individual feature importance

Explore the top-4 important features that impact your model's predictions (e.g., global explanations). Use the slider to show ascending feature importances. All values feature importances are shown side-by-side and can be toggled off by selecting the value in the legend. Click on any of the features in the graph to see a dependency plot below for that feature.

Top 4 features by their importance



Sort by cohort: All data

Chart type: Bar

Class importance: Weight

Average of absolute

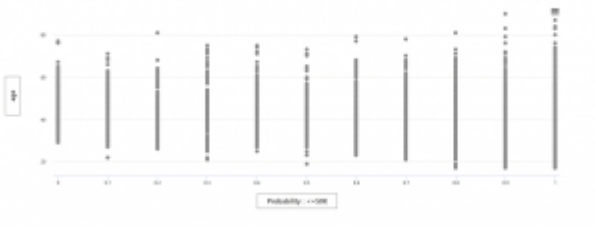
Select a feature to show its dependence plot

Show dependence plot for: Select feature

Select a dataset cohort: All data

Counterfactuals

What if allow you to perturb features for any input and observe how the model's prediction changes. You can perturb features manually or specify the desired prediction (e.g., class) for a dataset to see a list of closest data points to the original input that would lead to the desired prediction. Also known as prediction counterfactuals, you can use them for exploring the relationships learned by the model, understanding important, necessary features for the model's predictions, or adding edge cases for the model. To start, choose input points from the data table or scatter plot.



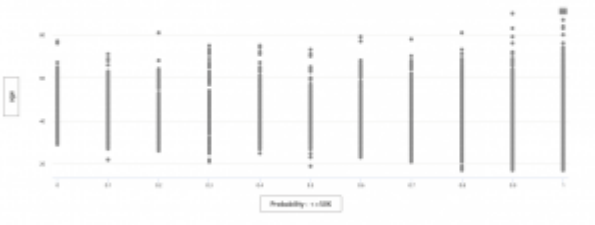
Selected dataset: All data

Current class: -10%

Select a data point to view local importance chart

Counterfactuals

What if allow you to perturb features for any input and observe how the model's prediction changes. You can perturb features manually or specify the desired prediction (e.g., class) for a dataset to see a list of closest data points to the original input that would lead to the desired prediction. Also known as prediction counterfactuals, you can use them for exploring the relationships learned by the model, understanding important, necessary features for the model's predictions, or adding edge cases for the model. To start, choose input points from the data table or scatter plot.



Selected dataset: All data

Current class: -10%

Select a data point to view local importance chart

Causal analysis

The overall causal effects are for selected data

Aggregate causal effects Individual causal effect Treatment policy

Causal analysis answers "what if" questions about how model predictions would have changed under different policy options, such as different pricing strategies for a product or an alternative treatment for a patient. Unlike model predictions that identify important correlation patterns, these tools help provide the most important causal features that directly affect your outcome of interest. These models identify the causal effect of one feature (typically referred to as a "treatment"), holding other confounding features constant. For best results, make sure that the full dataset contains all available features that may correlate with the outcome as confounders.

Select aggregate causal effect of each treatment with 95% confidence interval

Why is it important to include confounding features?

Feature	Effect estimate	Standard error	Z score	P-value	Confidence interval	Confidence interval support
education(bachelors)	0.027e+1	1.880e-1	0.000e+0	0.070e+1	0.000e+0	0.000e+0
education(post-grad)	0.770e+1	0.100e-1	0.070e+1	0.000e+0	0.000e+0	0.000e+0
education(masters)	0.460e+1	0.070e-1	0.000e+0	0.000e+0	0.000e+0	0.000e+0
education(high school)	0.000e+0	0.000e-1	0.000e+0	0.000e+0	0.000e+0	0.000e+0
education(some college)	0.000e+0	0.000e-1	0.000e+0	0.000e+0	0.000e+0	0.000e+0
education(associate)	0.000e+0	0.000e-1	0.000e+0	0.000e+0	0.000e+0	0.000e+0
education(bachelors)	0.000e+0	0.000e-1	0.000e+0	0.000e+0	0.000e+0	0.000e+0
education(masters)	0.000e+0	0.000e-1	0.000e+0	0.000e+0	0.000e+0	0.000e+0
education(post-grad)	0.000e+0	0.000e-1	0.000e+0	0.000e+0	0.000e+0	0.000e+0
education(some college)	0.000e+0	0.000e-1	0.000e+0	0.000e+0	0.000e+0	0.000e+0
education(associate)	0.000e+0	0.000e-1	0.000e+0	0.000e+0	0.000e+0	0.000e+0
education(bachelors)	0.000e+0	0.000e-1	0.000e+0	0.000e+0	0.000e+0	0.000e+0
education(masters)	0.000e+0	0.000e-1	0.000e+0	0.000e+0	0.000e+0	0.000e+0
education(post-grad)	0.000e+0	0.000e-1	0.000e+0	0.000e+0	0.000e+0	0.000e+0
education(some college)	0.000e+0	0.000e-1	0.000e+0	0.000e+0	0.000e+0	0.000e+0
education(associate)	0.000e+0	0.000e-1	0.000e+0	0.000e+0	0.000e+0	0.000e+0
education(bachelors)	0.000e+0	0.000e-1	0.000e+0	0.000e+0	0.000e+0	0.000e+0
education(masters)	0.000e+0	0.000e-1	0.000e+0	0.000e+0	0.000e+0	0.000e+0
education(post-grad)	0.000e+0	0.000e-1	0.000e+0	0.000e+0	0.000e+0	0.000e+0
education(some college)	0.000e+0	0.000e-1	0.000e+0	0.000e+0	0.000e+0	0.000e+0
education(associate)	0.000e+0	0.000e-1	0.000e+0	0.000e+0	0.000e+0	0.000e+0

Confidence intervals: On average in this sample, increasing this feature by 1 unit will cause the probability of class label "10%" to increase by 0 units.

Binary treatments: On average in this sample, turning on this feature will cause the probability of class label "10%" to increase by 0 units.

A linear for logistic regression $P(Y=1)$ is fitted with fit to predict a binary (0, 1) and a least-square regression $\beta(X)$ is computed with fit to predict $E(Y)$ from X . The causal effect can be viewed as the average contribution of the individual feature X_j to the log-probability ratio of the two predicted levels. Learn more about Double Machine Learning here.

Global cohort: All data (default) Switch cohort New cohort

Error analysis

See map [View map](#)

With this feature map you can focus on specific interactional feature filters and compare disaggregated error rates. Start with low-dimension features to compare.

Cohort: All data

Cells Error coverage Error rate

- 100.00% 15.88%

Select errors:

Rows: Position 1
 Columns: Feature 2

Select low features by using the dropdown below. You can double and filter your data along low-dimensions.

Basic information
 All data
 All data (2 filters)
Instances in global cohort
 Total: 2000
 Correct: 4206
 Incorrect: 794
Instances in the selected cohort
 Total: 2000
 Correct: 4206
 Incorrect: 794
Prediction path (filters)

Model overview

Evaluate the performance of your model by exploring the distribution of your predictor values and the values of your model performance metrics. Use the "Global cohort" tab to investigate your model's testing at a comparative analysis of its performance across different groups to easily understand cohort's. Use the "Feature cohort" to investigate your model by looking at a comparative analysis of its performance across sensitive/non-sensitive feature subgroups (e.g. performance across different genders, income levels).

Dataset cohorts: **Feature cohorts:**

Metrics: Accuracy score, F1 score, precision rate, F2, ... Help me choose metrics

Cohort	Sample size	Accuracy score	F1 score	F2 score	Selection size
All data	2,000	0.942	0.971	0.943	0.200

Probability distribution: **Metrics visualizations:** **Confusion matrix:**

See split chart

Data analysis

Table view Chart view

View the dataset in a table format for all features and rows.

Index True? Predicted? age workclass bkg education education-num sex

Correct predictions (4206)
 Incorrect predictions (794)

id	True?	Predicted?	age	workclass	bkg	education	education-num	sex
0	True	True	41	Non-emp	20504	Graduate	16	Male
10	True	True	34	Non-emp	20504	Some college	16	Male
11	True	True	41	Non-emp	19963	HS grad	9	Male
12	True	True	42	Non-emp	17562	Prof school	12	Male
22	True	True	41	Self-emp	17522	Assoc degree	12	Male
36	True	True	41	Non-emp	19971	HS grad	9	Male
44	True	True	39	Non-emp	20523	Assoc deg	12	Male
52	True	True	42	Non-emp	17524	No deg	8	Male

AI Knowledge