



# 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=5, 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.

Download

```
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.
```

## **Output**



### Data analysis

Table view **Chart view**

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

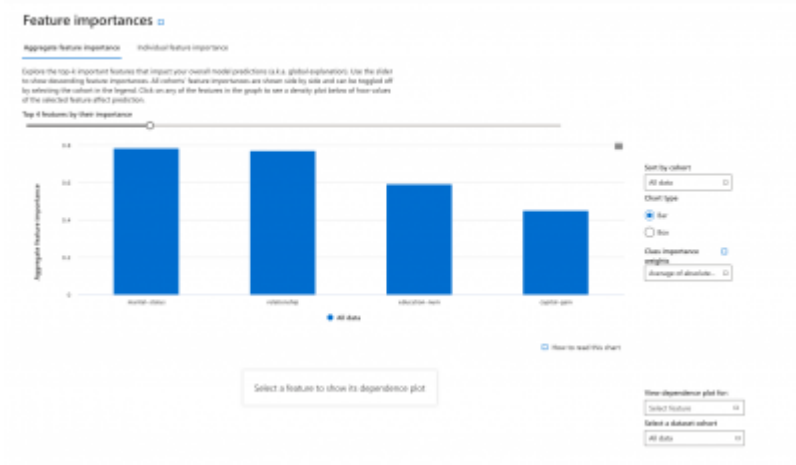
Index	TrueY	PredictedY	age	workclass	salary	education	education-num	sex
5	++00	++00	41	Private	20704	Bachelor	16	Male
10	++00	++00	34	Private	20004	Some college	16	Male
11	++00	++00	47	Private	40004	HS grad	9	Male
13	++00	++00	42	Private	17504	Prof school	10	Male
22	++00	++00	47	Local gov	17000	Assoc voc	12	Male
34	++00	++00	31	Private	16011	HS grad	9	Male
44	++00	++00	38	Private	20014	Assoc voc	11	Male
52	++00	++00	43	Private	12004	Tech sch	8	Male

### Data analysis

Table view **Chart view**

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

Index	TrueY	PredictedY	age	workclass	salary	education	education-num	sex
5	++00	++00	41	Private	20704	Bachelor	16	Male
10	++00	++00	34	Private	20004	Some college	16	Male
11	++00	++00	47	Private	40004	HS grad	9	Male
13	++00	++00	42	Private	17504	Prof school	10	Male
22	++00	++00	47	Local gov	17000	Assoc voc	12	Male
34	++00	++00	31	Private	16011	HS grad	9	Male
44	++00	++00	38	Private	20014	Assoc voc	11	Male
52	++00	++00	43	Private	12004	Tech sch	8	Male





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

### Error analysis

Tree map  Heat map

With the heat map you can focus on specific interactions between filters and compare disaggregated error rates. Start with low-impact features to compare.

Cohort: All data

Cells:  Error coverage:  100.00%  Error rate:  15.88%

Select metrics:

Select one feature by using the dropdown. Below, you can filter and filter your data along two dimensions.

Row: Position 1

Column: Feature 2

Save as a new cohort

Basic information  
All data  
All data (2 filters)

Instance in global cohort  
Total: 1000  
Correct: 836  
Incorrect: 164

Instance in the selected cohort  
Total: 1000  
Correct: 836  
Incorrect: 164

Prediction path (filter)

---

### 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 pre-built or newly created cohort filters. Use the "Feature cohort" to investigate your model by testing at a comparative analysis of its performance across categorical, high cardinality feature subgroups. (e.g. performance across different genders, income levels).

Tabular cohorts  Feature cohorts

Metrics:  Accuracy score, false positive rate, false negative rate, F1  Reply the chosen metrics

Cohort	Sample size	Accuracy score	False positive rate	False negative rate	Information
All data	1,000	0.842	0.075	0.043	0.936

Probability distribution  Metrics visualizations  Confusion matrix

See split chart

0 0.05 0.1 0.15 0.2 0.25 0.3 0.35 0.4 0.45 0.5 0.55 0.6 0.65 0.7 0.75 0.8 0.85 0.9 0.95 1

Probability: 0.500

---

### Data analysis

Table view  Chart view

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

ID	gender	marital	age	workclass	fnlwgt	education	education-num	sex
0	Female	Married	41	Never	20704	Bachelors	16	Male
10	Male	Married	34	Never	20884	Some-college	10	Male
11	Male	Married	47	Never	18863	HS grad	9	Male
12	Male	Married	42	Never	17542	Prof school	10	Male
22	Male	Married	47	Self emp	71020	Assoc-voc	10	Male
36	Male	Married	31	Never	16671	HS grad	9	Male
47	Male	Married	33	Never	20023	Assoc-voc	10	Male
53	Male	Married	33	Never	13074	No high school	8	Male

AI Knowledge