

Hospital Records for Data Cleaning (Medium)

NUDRAT ABBAS - UPDATED 9 DAYS AGO

28

Code Download

clinical terminology with missing values, date anomalies, and inconsistent label

Data Card Code (3) Discussion (0) Suggestions (0)

About Dataset

This dataset is designed for intermediate learners who want hands-on experience cleaning healthcare-style data.

Patient records are synthetic and anonymized, while diagnoses are based on real clinical terminology inspired by public biomedical vocabularies (e.g., MeSH).

The dataset intentionally contains logical inconsistencies and missing fields commonly found in healthcare data systems.

Usability 10.00

License CC0: Public Domain

Expected update frequency Annually

We will use Azure ML Designer to develop a model that will help us with this dataset for hospitals. In this case we will develop a model to predict the cases in a month given the diagnosis, day and month.

We need to start or create a Compute instance. You need to go to Azure ML Workspace and the select Compute option, then turn on or create your instance.

Choose from a selection of CPU or GPU instances preconfigured with popular tools such as VS Code, JupyterLab, Jupyter, and RStudio, ML packages, deep learning frameworks, and more. [about compute instances](#)

+ New Refresh Start Stop Restart Schedule and idle shutdown Delete Reset view

| Name | State | Idle shutdown | Applications |
|------------------------|---------|---------------|--------------------------------------|
| samplecomputeinstance | Stopped | 1 hour | JupyterLab Jupyter VS Code (Web) ... |
| simplecomputeinstance2 | Stopped | 1 hour | JupyterLab Jupyter VS Code (Web) ... |

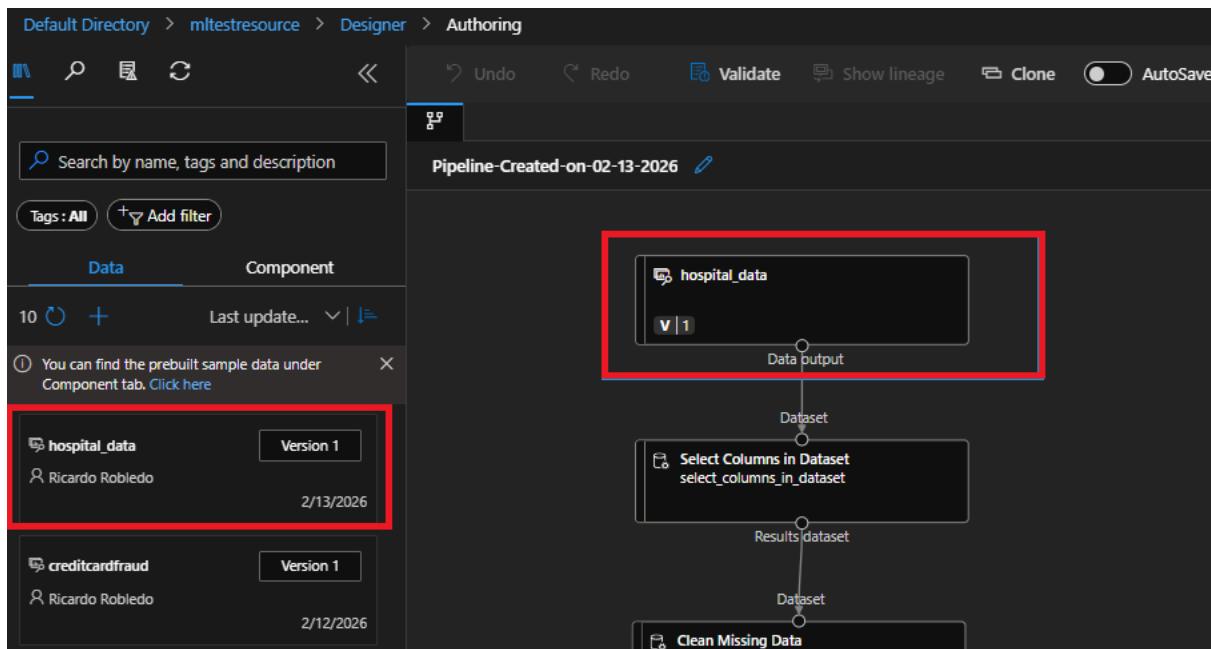
Then you need to have a dataset, for that you need to have a storage account and upload the dataset in a container instance, after that create a data asset with that file.

The screenshot shows the Azure Machine Learning studio interface. On the left, there's a navigation sidebar with sections like 'All workspaces', 'Home', 'Model catalog', 'Authoring' (which is expanded to show 'Notebooks', 'Automated ML', 'Designer', and 'Prompt flow'), 'Assets', and 'Data'. The 'Data' section is currently selected. The main area displays the details of a dataset named 'hospital_data'. The top bar shows the path 'Default Directory > mltestresource > Data > hospital_data' and the version 'Version: 1 (latest)'. Below the path are tabs for 'Details', 'Consume', 'Explore', 'Models', and 'Jobs', with 'Details' being the active tab. Under the 'Details' tab, there are buttons for 'New version', 'Refresh', 'Generate profile', and 'Archive'. The 'Attributes' section lists the type as 'Table (mltable)', dataset type as 'Tabular', and creator as 'Ricardo Robledo'. There are also sections for 'Tags' (no data), 'Description' (click edit), and 'Data source'.

Then we go to Designer and create a template to create our model. In this case I have a pipeline that I will explain.

The screenshot shows the 'Designer' interface in the Azure ML studio. The top navigation bar includes 'Default Directory', 'mltestresource', 'Designer', 'Authoring', 'Pipeline interface', 'Save', and 'Configure & Submit'. The main workspace is titled 'Pipeline-Created-on-02-13-2026'. On the left, there's a sidebar with a search bar, a 'Tags: All' filter, and a list of components categorized under 'Data' and 'Component'. The 'Component' tab is selected. The list includes 'Sample data (16)', 'Data Transformation (19)', 'Computer Vision (6)', 'Model Scoring & Evaluation (6)', 'Machine Learning Algorithms (19)', 'Text Analytics (7)', 'Python Language (2)', 'Data Input and Output (3)', 'Recommendation (5)', 'R Language (1)', and 'Feature Selection (2)'. The main workspace displays a complex data pipeline diagram with various components connected by arrows, representing a machine learning workflow from data input to final results.

Go to data and select the dataset.



Ok, it is important to know which columns we will use, in my case I used Azure Data Explorer and Kusto to explore data, we will make a model that predicts the total count of cases depending on the diagnosis.

The screenshot shows the Azure Data Explorer interface. At the top, there's a toolbar with 'Run', 'Recall', 'KQL tools', and a database dropdown set to 'MyFreeCluster/MyDatabase'. Below the toolbar is a code editor window displaying a Kusto query:

```

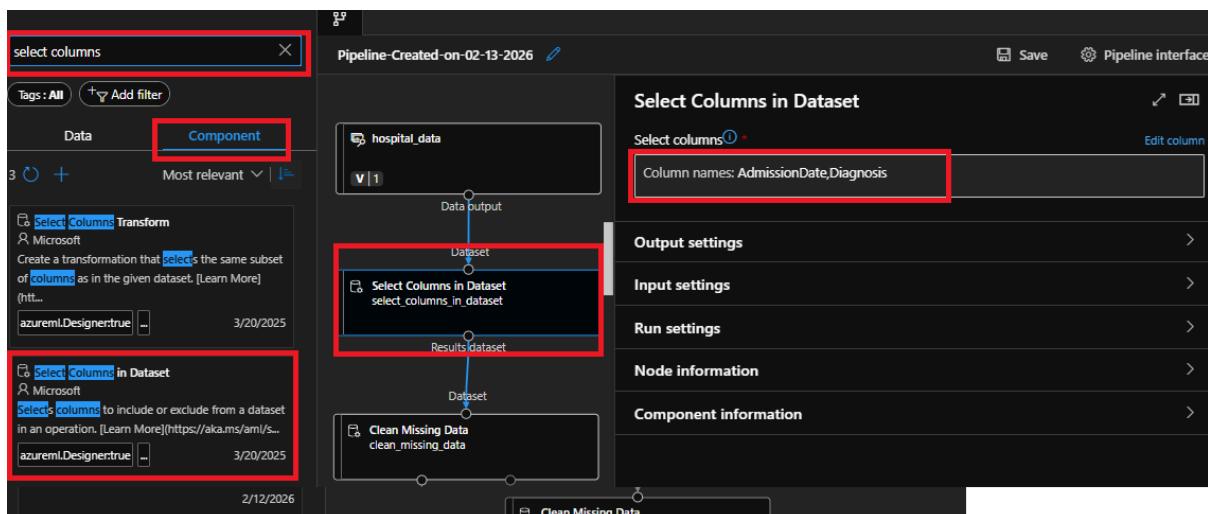
1 hospitaltesttable
2 | extend Year = toint(format_datetime(AdmissionDate, "yyyy")),
3 | | Month = toint(format_datetime(AdmissionDate, "MM"))
4 | summarize Cases = count()
5 | | by Diagnosis, Year, Month

```

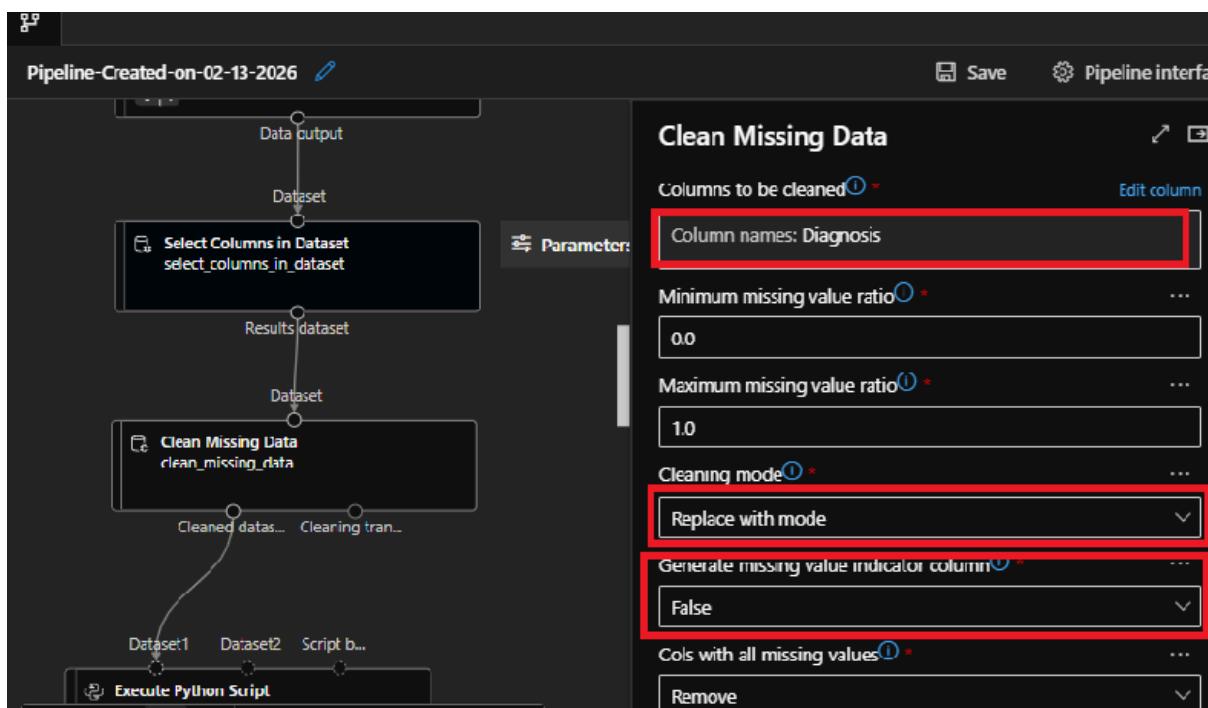
Below the code editor is a table titled 'Table 1'. The table has columns: Diagnosis, Year, Month, and Cases. The data is as follows:

| Diagnosis | Year | Month | Cases |
|-----------------------|-------|-------|-------|
| Myocardial Infarction | 2,024 | 3 | 11 |
| Pneumonia | 2,024 | 8 | 8 |
| Influenza | 2,024 | 11 | 12 |
| Acute Bronchitis | 2,025 | 7 | 14 |
| Type 2 Diabetes | 2,023 | 8 | 9 |
| Gastroenteritis | 2,024 | 8 | 10 |

Now we must go to select components and drag Select Columns in Dataset, after that select the next columns.



Now we need to use the component Clean Missing Data. Select column Diagnosis, replace with mode and not generate another column.



The next is to put a Execute Python Script module, the code is the next:

```
import pandas as pd

def azureml_main(dataframe1=None, dataframe2=None):

    # Copiar solo columnas necesarias
    df = dataframe1[["Diagnosis", "AdmissionDate"]].copy()
```

```

# Convertir a datetime
df["AdmissionDate"] = pd.to_datetime(df["AdmissionDate"])

# Crear Year y Month
df["Year"] = df["AdmissionDate"].dt.year
df["Month"] = df["AdmissionDate"].dt.month

# Agrupar y contar
result = (
    df.groupby(["Diagnosis", "Year", "Month"])
    .size()
    .reset_index(name="Cases")
    .sort_values(["Year", "Month", "Cases"],
                ascending=[True, True, False])
)

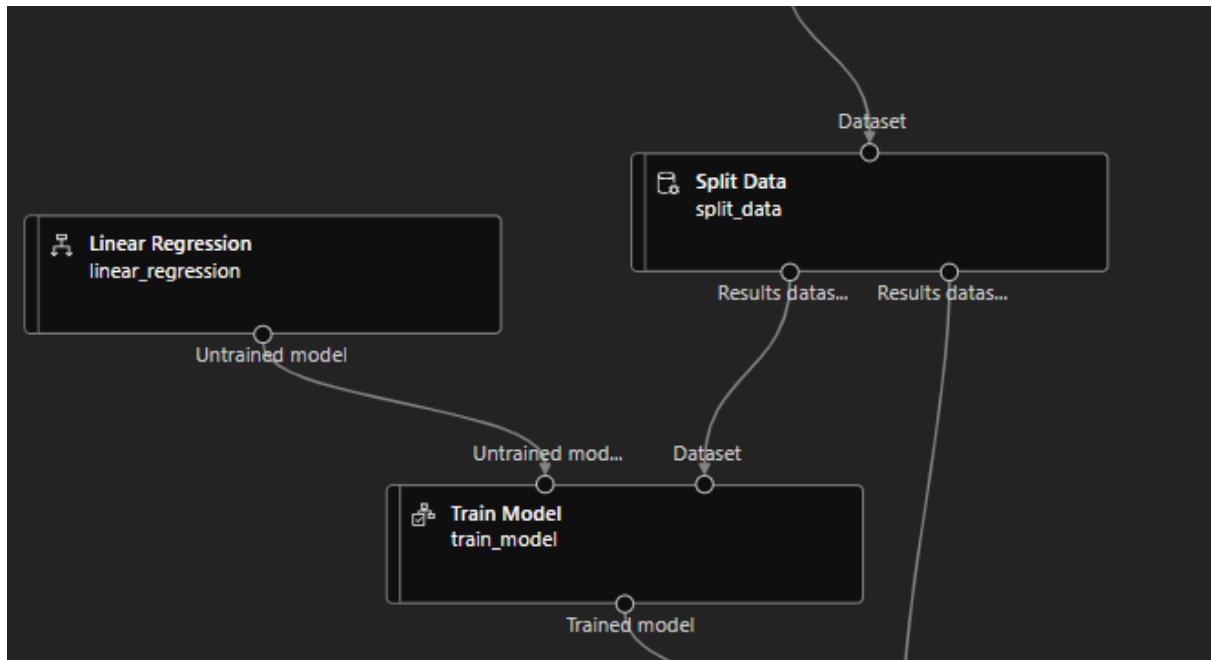
# Devolver SOLO las columnas deseadas explicitamente
result = result[["Diagnosis", "Year", "Month", "Cases"]]

return result,

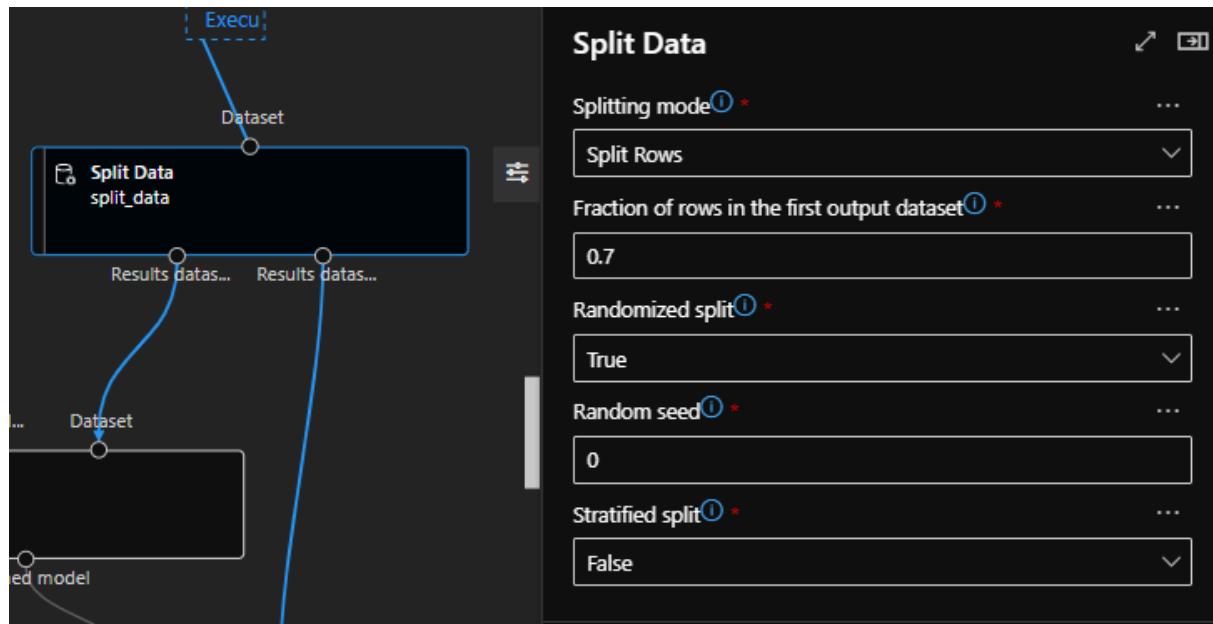
```

In a brief summary it only group the diagnosis by month and year, then we only return the columns that we will use to predict the Cases.

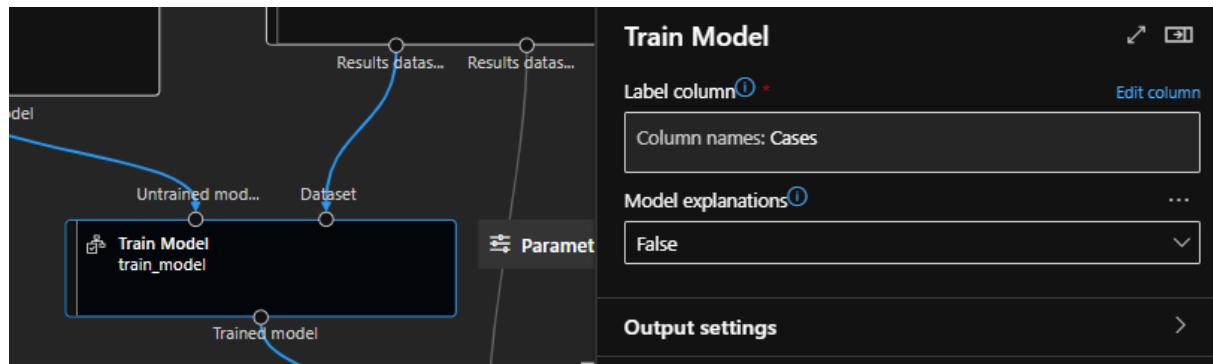
Ok, we will use 3 modules, they are: Split Data, Liner Regression and Train Model.



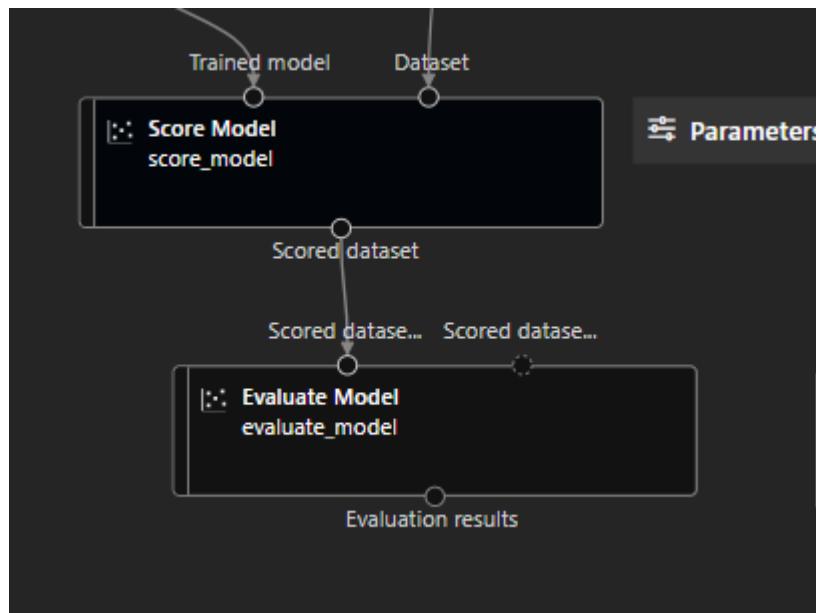
These are the settings for Split Data. Like we have temporal data, we don't need randomized split neither stratified split.



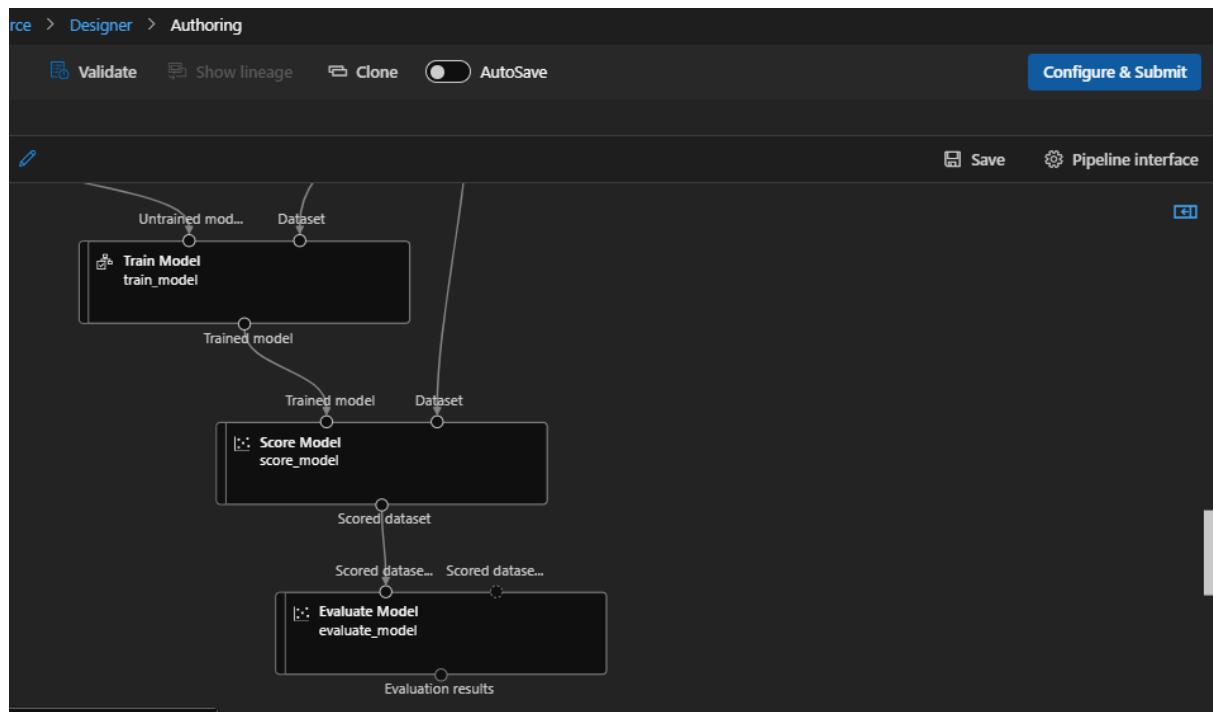
Train Model is very simple, only select the target that is Cases.



After that we need only 2 modules that are Score Model and Evaluate Model.



Very well, we only need to execute our pipeline, to that push Configure & Submit.



Give it a name to the experiment.

Set up pipeline job

1 Basics 2 Inputs & outputs 3 Runtime settings 4 Review + Submit

Basics

Experiment name
 Select existing Create new
New experiment name *
test

Job display name
Pipeline-Created-on-02-13-2026

Job description
Pipeline created on 20260213

Job tags
Name : Value

Select Compute instance, and Review + Submit.

Set up pipeline job

1 Basics 2 Inputs & outputs 3 Runtime settings 4 Review + Submit

Runtime settings

Default compute ⓘ
The pipeline compute target samplecomputeinstance is invalid.

Select compute type
Compute instance

Select Azure ML compute instance
samplecomputeinstance, Stopped, Standard_DS11_v2, 2 vCPUs (cores), 14 GB, ...

Identity

Default datastore ⓘ
Select datastore *
workspacesampled100

Advanced settings
 Continue on step failure ⓘ

after that the pipeline will be created, so after some minutes we can go to Jobs in the menu and Select pipeline.

The screenshot shows the Databricks interface with the sidebar navigation open. The 'Jobs' option is selected under the 'Jobs' section. The main content area displays a list of jobs, with one job highlighted by a red box. The highlighted job is titled 'Pipeline-Created-on-02-13-202 (8)' and has a status of 'Completed' with a green checkmark. The table includes columns for Display name, Parent job name, Status, Created on, and Duration.

| Display name (7 visualized) | Parent job name | Status | Created on | Duration |
|-----------------------------------|-----------------|-----------|-----------------------|----------|
| Pipeline-Created-on-02-13-202 (8) | | Completed | Feb 13, 2026 9:33 AM | 2m 7s |
| Pipeline-Created-on-02-13-202 (8) | | Completed | Feb 13, 2026 9:28 AM | 2m 56s |
| Pipeline-Created-on-02-13-202 (8) | | Completed | Feb 13, 2026 9:15 AM | 4m 0s |
| Pipeline-Created-on-02-11-202 (8) | | Completed | Feb 13, 2026 9:13 AM | 3m 5s |
| epic_jicama_c2ldq2xomm (7) | | Completed | Feb 12, 2026 8:29 AM | 17m 28s |
| Pipeline-Created-on-02-11-202 (6) | | Failed | Feb 11, 2026 8:20 AM | 3m 34s |
| Pipeline-Created-on-01-29-202 (1) | | Completed | Jan 31, 2026 10:42 AM | 55s |

OK!, this is good, all the modules have executed successfully.

Pipeline-Created-on-02-13-2026

Completed

Share Add to compare Job overview

Imported data

Select Columns by Element

Clean Missing Data

Drop Nulls

Random Forest - Model

Split Data

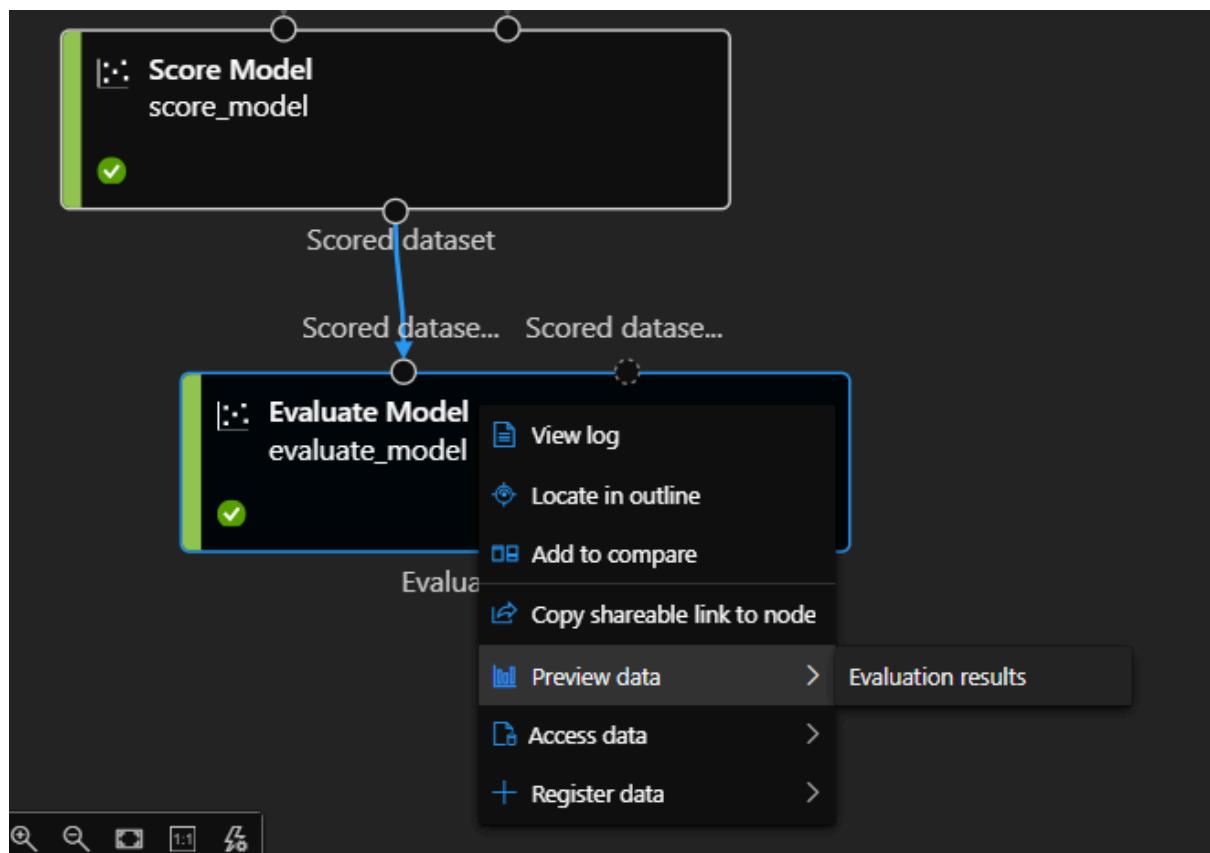
Train Model

Test Model

Score Model

Navigator 43% Show lineage Schedule Show compare list Cancel

Find the Evaluate Model module and do right click and select the next option..



These are our results.

| Evaluation_results | |
|------------------------------|-------------------------|
| Rows | Columns |
| 1 | 5 |
| Mean_Absolute_Error | Root_Mean_Squared_Error |
| 2.0259 | 2.806342 |
| Relative_Squared_Error | Relative_Absolute_Error |
| 0.299339 | 0.471088 |
| Coefficient_of_Determination | |
| 0.700661 | |