

Credit card Fraud data

Fraud Detection credit card

Data Card Code (5) Discussion (0) Suggestions (0)

About Dataset

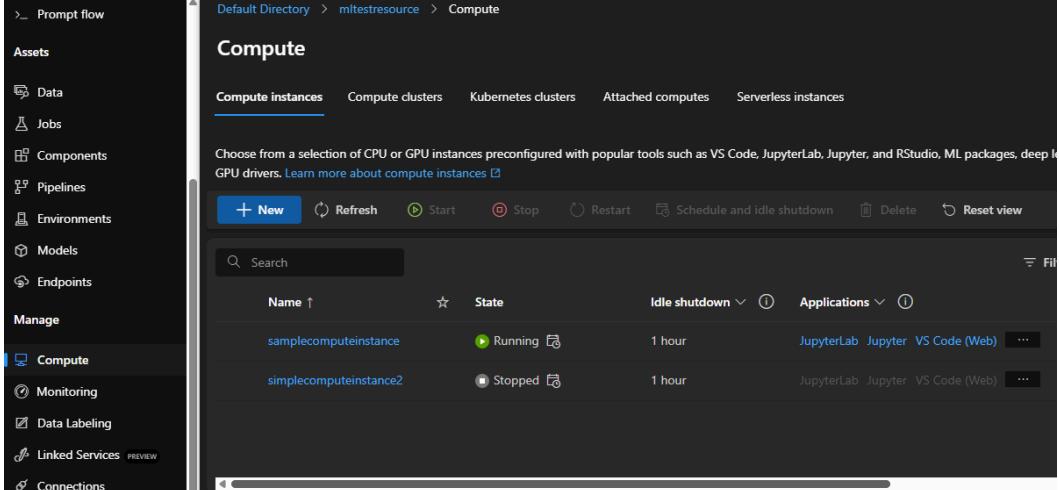
This dataset contains credit card transaction records used to identify fraudulent transactions. The primary goal is to build and evaluate machine learning models capable of accurately distinguishing between legitimate and fraudulent activities.

The dataset includes transactions made by credit card users, where each transaction is labeled as fraud (1) or non-fraud (0)

Usability 10.00
License CC0: Public Domain
Expected update frequency Quarterly
Tags

We will solve this problem creating a ML model to predict credit card fraud data, this dataset is disbalanced.

First you need to create and execute a compute instance.



Default Directory > mltestresource > Compute

Compute

Compute instances Compute clusters Kubernetes clusters Attached computes Serverless instances

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 GPU drivers. [Learn more about compute instances](#)

Name	State	Idle shutdown	Applications	⋮
samplecomputeinstance	Running	1 hour	JupyterLab Jupyter VS Code (Web)	⋮
simplecomputeinstance2	Stopped	1 hour	JupyterLab Jupyter VS Code (Web)	⋮

We create a datastore and put our dataset.

Default Directory > miltestresource > Data > third datastore

third datastore

Overview Browse

Create data asset Refresh Update authentication Set as default datastore Unregister

General

Datastore name: third datastore
Datastore type: Azure Blob Storage
Created by: Ricardo Robledo
Subscription ID: --
Resource group name: --
Protocol: https

Browse preview

Name Created on

Name	Created on
UI	--
managed-dataset	--
pipeline-outputs	--
flightdelaydata.csv	Jan 31, 2026 10:29 AM
fraudTest.csv	Feb 12, 2026 7:16 AM
scoredataset.csv	Feb 2, 2026 5:50 AM

Then create the Train automatically option.

Submit an Automated ML job

1 Training method
2 Basic settings
3 Task type & data
4 Task settings
5 Compute
6 Review

How do you want to train your model?

Choose the method of training you would like to use.

Train automatically

Submit an Automated ML job to train a model without writing a single line of code.

Run a custom training script

Submit a command job to train your model using your own code.

Perform hyperparameter tuning

Submit a sweep job that performs hyperparameter tuning using random, grid or bayesian sampling algorithms.

Back Start configuring job

We put a name to the job and the experiment.

Submit an Automated ML job

Training method

Basic settings

Task type & data

Task settings

Compute

Review

Job name * ⓘ

Experiment name *

Select existing Create new

New experiment name *

Description

Tags

Name	:	Value
------	---	-------

Add

Back Next

Select the task type, in this case classification. Then push create to create a dataset.

Submit an Automated ML job

Training method

Basic settings

Task type & data

Classification

Select data

+ Create Refresh Show supported data assets only Reset view

Search Filter

Name	Type	Created on	Modified
------	------	------------	----------

Back Next

Select Azure Storage from our storage account.

Create data asset

Choose a source for your data asset
Choose the data source you want to create your asset from. A data source can be from a local storage location on your computer, from an attached datastore, from Azure storage, or from a publicly available web location.

From Azure storage

Create a data asset from registered data storage services including Azure Blob Storage, Azure file share, and Azure Data Lake.

From local files

Create a data asset by uploading files from your local drive.

From SQL databases

Create a dataset from Azure SQL database and Azure PostGreSQL database.

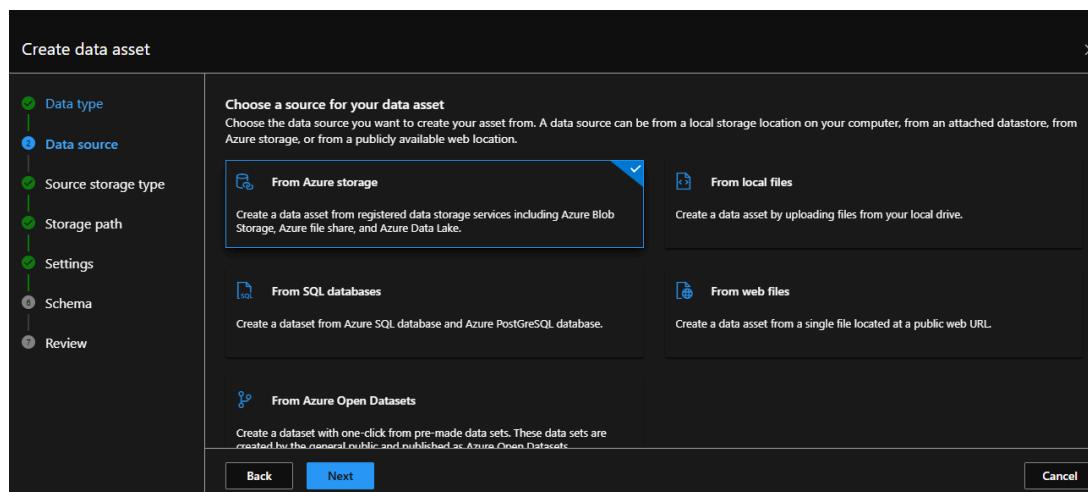
From web files

Create a data asset from a single file located at a public web URL.

From Azure Open Datasets

Create a dataset with one-click from pre-made data sets. These data sets are created by the general public and published as [Azure Open Datasets](#).

Back Next Cancel



Select datastore

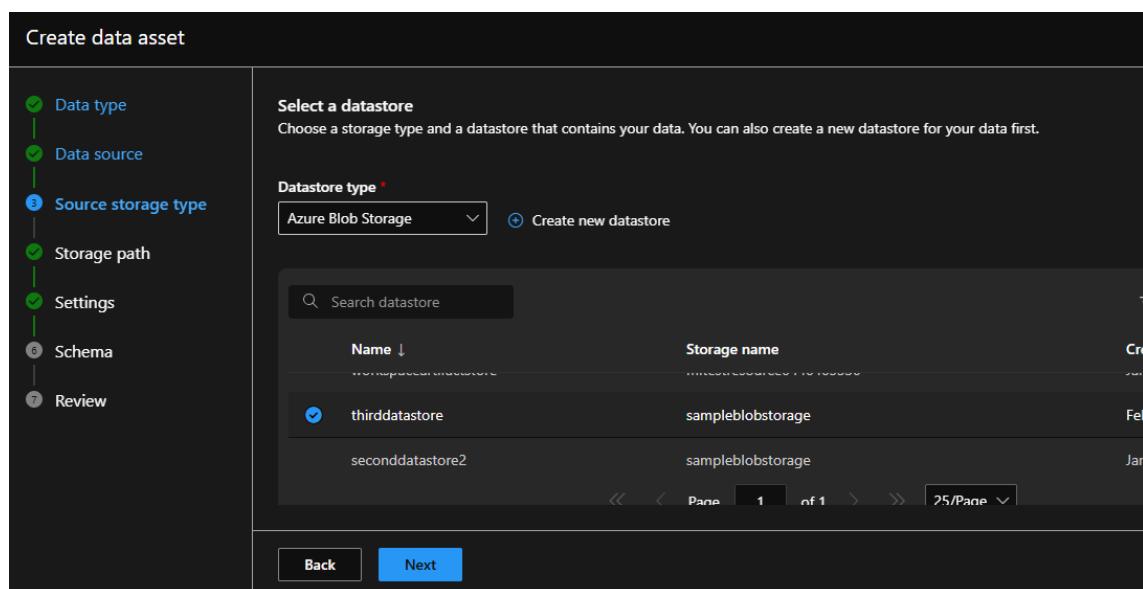
Create data asset

Select a datastore
Choose a storage type and a datastore that contains your data. You can also create a new datastore for your data first.

Datastore type *
Azure Blob Storage [Create new datastore](#)

Name	Storage name	Created
thirddatastore	sampleblobstorage	Feb
seconddatastore2	sampleblobstorage	Jan

Back Next



Confirm data

Create data asset

- Data type
- Data source
- Source storage type
- Storage path
- Settings
- Schema
- Review

Settings
These settings determine how the data is parsed. The initial settings are automatically detected; you can change them as needed to reparse the data.

File format	Delimiter	Example	Encoding
Delimited	Comma	Field1,Field2,Field3	UTF-8

Column headers	Skip rows
All files have same headers	None

Dataset contains multi-line data ⓘ

ⓘ Note: Processing tabular files with multi-line data is slower because multiple CPU cores cannot be used to ingest the data in parallel. Checking this option may result in slower Data previewing times.

sn	trans_d...	cc_num	merchant	category	amt	first	last	gender	street	city	state	zip
0	2020-06...	2,291,16...	fraud_Ki...	persona...	2.86	Jeff	Elliott	M	351 Dari...	Columbia	SC	2920
1	2020-06...	3,573,03...	fraud_S...	persona...	29.84	Joanne	Williams	F	3638 M...	Altonah	UT	8400
2	2020-06...	3,598,22...	fraud_S...	health fi...	41.28	Ashley	Lopez	F	9333 Valu...	Bellmore	NY	1171

Back **Next** **Review** **Cancel**

Include and change values needed. In this case we use: *trans_date_trans_time, category, amt, gender, city, state, zip, lat, long, city_pop, job, dob, unix_time, merch_lat, merch_long, is_fraud*.

delete: *trans_num, street, first, last, cc_num, sn, merchant*.

Create data asset

- Data type
- Data source
- Source storage type
- Storage path
- Settings
- Schema
- Review

Schema
Column types are auto-detected based on the initial subset of the data and can be updated here. Values not aligning with the specified column type will fail conversion and would be either null-filled or replaced with error value. Any conversions preview errors are non-blocking and you can proceed.

Include	Column name	Type	Example values	Date format <small>ⓘ</small>	Properties <small>ⓘ</small>
<input checked="" type="radio"/>	Path	String		Not applicable to s...	Not applicable t...
<input checked="" type="radio"/>	sn	Integer	0, 1, 2	Not applicable to s...	Not applicable t...
<input checked="" type="radio"/>	trans_date_trans_time	Date	2020-06-21 12:14:00, 202...	%d-%m-%Y %H:%M	<input type="radio"/> None
<input checked="" type="radio"/>	cc_num	Decimal (dot ',')	229116000000000.3573...	Not applicable to s...	Not applicable t...

Back **Next** **Cancel**

Verify and create dataset.

Create data asset

Review
Review the settings for your data asset and make any changes as needed.

Data type

Name
creditcardfraud

Description
--

Type
tabular

Data source

Type
AzureStorage

Storage

Datashare type
--

Schema

trans_date_trans_time	Date
category	String
amt	Decimal
gender	String
city	String

(showing 5 of 24 columns)

Back Create Cancel

We'll use `is_fraud` like our target column and `AveragePrecisionScoreWeighted` like metric.

Default Directory > mltestresource > Training job

Submit an Automated ML job

Additional configuration

Primary metric ⓘ

AveragePrecisionScoreWeighted

Explain best model ⓘ

Enable ensemble stacking ⓘ

Use all supported models

Blocked models ⓘ

A list of models that Automated ML will not use during training.

Positive class label ⓘ

Positive class label

Back Next Save Close

Select your compute instance.

Default Directory > mltestresource > Training job

Submit an Automated ML job

Training method
Basic settings
Task type & data
Task settings
Compute
Review

Compute

Select and configure the compute resource for executing your training job.

Select compute type

Compute instance

Select Azure ML compute instance*

samplecomputeinstance, Running, Standard_DS11_v2, 2 vCPUs (cores), 14 GB, 28 GB (storage)

samplecomputeinstance - Running
Standard_DS11_v2
2 vCPUs (cores), 14 GB, 28 GB (storage), \$0.18/hr

simplecomputeinstance2 - Stopped
Standard_A1_v2
1 vCPU (core), 2 GB, 10 GB (storage), \$0.04/hr

Back Next

Review and create.

Default Directory > mltestresource > Training job

Submit an Automated ML job

Training method
Basic settings
Task type & data
Task settings
Compute
Review

Review

Review or make changes to your job before submission.

Basic settings

Name: epic_jicama_c2ldq2xmm
Experiment name: test
Timeout (hours): --

Task type & data

Task type: Classification
Data: creditcardfraud

Task settings

Target column: is_fraud
Max trials: 3
Max concurrent: 1
Max nodes: 1
Metric score: F1
Experiment timeout: 1000
Iteration time: 1000
Enable deep learning: No

Back Submit training job Cancel

After this is completed we can verify the results.

Default Directory > mltestresource > Jobs > test > epic_jicama_c2ldq2xxmm

epic_jicama_c2ldq2xxmm Completed

Overview Data guardrails Models + child jobs Outputs + logs Child jobs

Refresh Edit and submit (preview) Register model Cancel Delete Compare

Properties	
Status	Completed
Created on	Feb 12, 2026 8:29 AM
Start time	Feb 12, 2026 8:29 AM
Duration	17m 28.31s
Compute duration	17m 28.31s

Inputs	
Input name:	training_data
Data asset:	creditcardfraud:1
Asset URI:	azureml:creditcardfraud:1

Outputs	
Output name:	best_model
Model:	azureml_epic_jicama_c2ldq2xxmm_1_output_mlflow_log_model_209276562:1
Asset URI:	azureml:azureml_epic_jicama_c2ldq2xxmm_1_output_mlflow_L...

These are the results.

Default Directory > mltestresource > Jobs > test > epic_jicama_c2ldq2xxmm

epic_jicama_c2ldq2xxmm Completed

Overview Data guardrails Models + child jobs Outputs + logs

Compute duration 17m 28.31s

Compute target samplecomputeinstance

Name epic_jicama_c2ldq2xxmm

Script name --

Created by Ricardo Robledo

Job type Automated ML

Experiment test

Run Metrics

Accuracy	0.99916
AUC macro	0.99820
AUC micro	0.99998
AUC weighted	0.99820
Average precision score macro	0.97032
Average precision score micro	0.99998
Average precision score weighted	0.99976
Balanced accuracy	0.90675
F1 score macro	0.94078

The screenshot shows a dark-themed user interface for managing machine learning jobs. On the left, a sidebar displays navigation paths: Default Directory > mltestresource > Jobs > test > epic_jicama_c2. Below this, the job name is listed as "epic_jicama_c2ldq2xxmm" with edit, star, and checkmark icons, indicating it is "Completed". A horizontal menu bar includes "Overview", "Data guardrails", "Models + child jobs", and "Outputs + logs", with "Overview" being the active tab. The main content area on the right contains two columns of job metrics. The left column lists properties like "Compute target" (samplecomputeinstance), "Name" (epic_jicama_c2ldq2xxmm), "Script name" (empty), "Created by" (Ricardo Robledo), "Job type" (Automated ML), "Experiment" (test), and "Arguments" (None). The right column lists performance metrics with their values: Log loss (0.0029130), Matthews correlation (0.88474), Norm macro recall (0.81351), Precision score macro (0.98113), Precision score micro (0.99916), Precision score weighted (0.99914), Recall score macro (0.90675), Recall score micro (0.99916), Recall score weighted (0.99916), and Weighted accuracy (0.99988). A "Close" button is located at the bottom right of the metrics panel.

Property	Value
Compute target	samplecomputeinstance
Name	epic_jicama_c2ldq2xxmm
Script name	--
Created by	Ricardo Robledo
Job type	Automated ML
Experiment	test
Arguments	None
Can all properties	Close
Log loss	0.0029130
Matthews correlation	0.88474
Norm macro recall	0.81351
Precision score macro	0.98113
Precision score micro	0.99916
Precision score weighted	0.99914
Recall score macro	0.90675
Recall score micro	0.99916
Recall score weighted	0.99916
Weighted accuracy	0.99988

That's all!, this is the model finished.