

```
# Lab: Train-Test Split with Census Income Dataset
# 0. Installation and Imports
```

```
!pip install ucimlrepo
```

```
from ucimlrepo import fetch_ucirepo
import pandas as pd
```

```
Collecting ucimlrepo
```

```
  Downloading ucimlrepo-0.0.7-py3-none-any.whl.metadata (5.5 kB)
Requirement already satisfied: pandas>=1.0.0 in /usr/local/lib/python3.12/dist-packages (from ucimlrepo) (2.2.2)
Requirement already satisfied: certifi>=2020.12.5 in /usr/local/lib/python3.12/dist-packages (from ucimlrepo) (2025.11.12)
Requirement already satisfied: numpy>=1.26.0 in /usr/local/lib/python3.12/dist-packages (from pandas>=1.0.0->ucimlrepo) (2.0.2)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.12/dist-packages (from pandas>=1.0.0->ucimlrepo) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.12/dist-packages (from pandas>=1.0.0->ucimlrepo) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.12/dist-packages (from pandas>=1.0.0->ucimlrepo) (2025.1)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from python-dateutil>=2.8.2->pandas) (1.17.0)
Downloading ucimlrepo-0.0.7-py3-none-any.whl (8.0 kB)
Installing collected packages: ucimlrepo
Successfully installed ucimlrepo-0.0.7
```

```
# Lab: Train-Test Split with Census Income Dataset
# 1. Data Loading and Metadata
```

```
# Fetch census income dataset
census_income = fetch_ucirepo(id=20)
```

```
# Extract features and targets as DataFrames
X = census_income.data.features
y = census_income.data.targets
```

```
# Display metadata and variable information
print("Metadata:\n", census_income.metadata)
print("\nVariable Information:\n", census_income.variables)
```

```
Metadata:
```

```
{'uci_id': 20, 'name': 'Census Income', 'repository_url': 'https://archive.ics.uci.edu/dataset/20/census+income', 'description': 'The Census Income Dataset is a subset of the 1994 US Census data. It contains information about the demographic characteristics of the population and their income levels. The dataset is divided into two parts: a training set and a test set. The training set is used to build a model, and the test set is used to evaluate the model's performance. The dataset is available in several formats, including CSV, JSON, and XML. The dataset is also available in a Python package, which can be installed using pip. The package provides a convenient way to load the data into a pandas DataFrame. The package also includes a function to fetch the data from the UCI repository. The package is available on PyPI under the name 'ucimlrepo'.
```

```
Variable Information:
```

	name	role	type	demographic	\
0	age	Feature	Integer	Age	
1	workclass	Feature	Categorical	Income	
2	fnlwgt	Feature	Integer	None	
3	education	Feature	Categorical	Education Level	
4	education-num	Feature	Integer	Education Level	
5	marital-status	Feature	Categorical	Other	
6	occupation	Feature	Categorical	Other	
7	relationship	Feature	Categorical	Other	
8	race	Feature	Categorical	Race	
9	sex	Feature	Binary	Sex	
10	capital-gain	Feature	Integer	None	
11	capital-loss	Feature	Integer	None	
12	hours-per-week	Feature	Integer	None	
13	native-country	Feature	Categorical	Other	
14	income	Target	Binary	Income	

  

	description	units	missing_values
0	N/A	None	no
1	Private, Self-emp-not-inc, Self-emp-inc, Feder...	None	yes
2	None	None	no
3	Bachelors, Some-college, 11th, HS-grad, Prof...	None	no
4	None	None	no
5	Married-civ-spouse, Divorced, Never-married, S...	None	no
6	Tech-support, Craft-repair, Other-service, Sal...	None	yes
7	Wife, Own-child, Husband, Not-in-family, Other...	None	no
8	White, Asian-Pac-Islander, Amer-Indian-Eskimo, ...	None	no
9	Female, Male.	None	no
10	None	None	no
11	None	None	no
12	None	None	no
13	United-States, Cambodia, England, Puerto-Rico, ...	None	yes
14	>50K, <=50K.	None	no

```
# Lab: Train-Test Split with Census Income Dataset
# 2. Train-Test Split
```

```
from sklearn.model_selection import train_test_split
```

```
# Split data (80% train, 20% test)
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
# Display lengths to confirm sizes
print("Train set size:", X_train.shape[0])
print("Test set size:", X_test.shape[0])
```

```
Train set size: 39073
Test set size: 9769
```

Start coding or [generate](#) with AI.