```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler


pip install ucimlrepo
```

```
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/python3.11/dist-packages (from ucimlrepo) (2.2.2)
Requirement already satisfied: certifi>=2020.12.5 in /usr/local/lib/python3.11/dist-packages (from ucimlrepo) (2025.8.3)
Requirement already satisfied: numpy>=1.23.2 in /usr/local/lib/python3.11/dist-packages (from pandas>=1.0.0->ucimlrepo) (2.0.2)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dist-packages (from pandas>=1.0.0->ucimlrepo) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas>=1.0.0->ucimlrepo) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas>=1.0.0->ucimlrepo) (2025.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.8.2->pandas>=1.0.0->ucimlrepo) (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
```

```
from ucimlrepo import fetch_ucirepo

# fetch dataset
breast_cancer_wisconsin_diagnostic = fetch_ucirepo(id=17)

# data (as pandas dataframes)
X = breast_cancer_wisconsin_diagnostic.data.features
y = breast_cancer_wisconsin_diagnostic.data.targets

# metadata
print(breast_cancer_wisconsin_diagnostic.metadata)

# variable information
print(breast_cancer_wisconsin_diagnostic.variables)
```

```
                name     role        type demographic default_target info
9      concave_points1  Feature  Continuous        None           None  None
10           symmetry1  Feature  Continuous        None           None  None
11  fractal_dimension1  Feature  Continuous        None           None  None
12             radius2  Feature  Continuous        None           None  None
13            texture2  Feature  Continuous        None           None  None
14          perimeter2  Feature  Continuous        None           None  None
15               area2  Feature  Continuous        None           None  None
16         smoothness2  Feature  Continuous        None           None  None
17        compactness2  Feature  Continuous        None           None  None
18          concavity2  Feature  Continuous        None           None  None
19     concave_points2  Feature  Continuous        None           None  None
20           symmetry2  Feature  Continuous        None           None  None
21  fractal_dimension2  Feature  Continuous        None           None  None
22             radius3  Feature  Continuous        None           None  None
```

```
0              no
1              no
2              no
3              no
4              no
5              no
6              no
7              no
8              no
9              no
10             no
11             no
12             no
13             no
14             no
15             no
16             no
17             no
18             no
19             no
20             no
21             no
22             no
23             no
24             no
25             no
26             no
27             no
28             no
29             no
30             no
31             no
```

```
import pandas as pd

# Define column names based on dataset documentation
columns = [
    "id", "clump_thickness", "uniformity_cell_size", "uniformity_cell_shape",
    "marginal_adhesion", "single_epithelial_cell_size", "bare_nuclei",
    "bland_chromatin", "normal_nucleoli", "mitoses", "class"
]

# Load dataset directly from UCI
url = "https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/breast-cancer-wisconsin.data"
df = pd.read_csv(url, names=columns)

# Preview
print(df.head())
```

```
        id  clump_thickness  uniformity_cell_size  uniformity_cell_shape  \
0  1000025                5                     1                      1
1  1002945                5                     4                      4
2  1015425                3                     1                      1
3  1016277                6                     8                      8
4  1017023                4                     1                      1

   marginal_adhesion  single_epithelial_cell_size bare_nuclei  \
0                  1                            2           1
1                  5                            7          10
2                  1                            2           2
3                  1                            3           4
4                  3                            2           1
```

	bland_chromatin	normal_nucleoli	mitoses	class
0	3	1	1	2
1	3	2	1	2
2	3	1	1	2
3	3	7	1	2
4	3	1	1	2

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
import joblib


# Step 1: Load dataset from UCI
url = "https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/breast-cancer-wisconsin.data"
columns = [
    "id", "clump_thickness", "uniformity_cell_size", "uniformity_cell_shape",
    "marginal_adhesion", "single_epithelial_cell_size", "bare_nuclei",
    "bland_chromatin", "normal_nucleoli", "mitoses", "class"
]
df = pd.read_csv(url, names=columns)


# Step 2: Replace '?' with NaN and convert columns to numeric
df.replace('?', pd.NA, inplace=True)
df = df.apply(pd.to_numeric, errors='coerce')


# Step 3: Drop rows with missing values
df.dropna(inplace=True)


# Step 4: Drop the 'id' column
df.drop(columns=["id"], inplace=True)


# Step 5: Separate features and target
X = df.drop(columns=["class"])
y = df["class"]


# Step 6: Standardize features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)


# Step 7: Train-test split (25% test size)
X_train, X_test, y_train, y_test = train_test_split(
    X_scaled, y, test_size=0.25, random_state=42
)


# Step 8: Save preprocessed data for reuse
joblib.dump((X_train, X_test, y_train, y_test), "preprocessed_data.pkl")

print("✅ Preprocessing complete. Data saved to 'preprocessed_data.pkl'")
```

```
✅ Preprocessing complete. Data saved to 'preprocessed_data.pkl'
```

```
import joblib
from sklearn.linear_model import LogisticRegression
```

```
from sklearn.metrics import accuracy_score, confusion_matrix


# Step 1: Load preprocessed data
X_train, X_test, y_train, y_test = joblib.load("preprocessed_data.pkl")


# Step 2: Initialize and train the model
model = LogisticRegression()
model.fit(X_train, y_train)
```

```
▾ LogisticRegression
LogisticRegression()
```

```
# Step 3: Make predictions
y_pred = model.predict(X_test)


# Step 4: Evaluate performance
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)


# Step 5: Display results
print("🔍 Logistic Regression Results")
print(f"Accuracy: {accuracy:.4f}")
print("Confusion Matrix:")
print(conf_matrix)
```

```
🔍 Logistic Regression Results
Accuracy: 0.9532
Confusion Matrix:
[[102   1]
 [  7  61]]
```

Start coding or generate with AI.