

```

import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix,
ConfusionMatrixDisplay
import matplotlib.pyplot as plt

# Load dataset
df = pd.read_csv("MLEX3.csv")

# Encode 'gender' column
le = LabelEncoder()
df['gender'] = le.fit_transform(df['gender']) # 'male' → 1, 'female' → 0 (usually)

# Feature and target selection
x = df[['age', 'gender', 'bmi', 'blood_pressure', 'cholesterol']] # Ensure column names are exact
y = df['condition']

# Scale features
scaler = StandardScaler()
xscale = scaler.fit_transform(x)

# Train-test split
xtr, xte, ytr, yte = train_test_split(xscale, y, test_size=0.2, random_state=42)

# Train logistic regression model
model = LogisticRegression()
model.fit(xtr, ytr)

# Predictions
ypr = model.predict(xte)
yprob = model.predict_proba(xte)[:, 1]

# Evaluation
print("Accuracy:", accuracy_score(yte, ypr))
print("Classification Report:\n", classification_report(yte, ypr, zero_division=1))

# Confusion matrix
cm = confusion_matrix(yte, ypr)
disp = ConfusionMatrixDisplay(confusion_matrix=cm)
disp.plot(cmap='Blues')
plt.title("Confusion Matrix")
plt.show()

# Predicting for a new patient: age=60, gender='male', bmi=27, blood_pressure=130,
cholesterol=200
# Encode gender as numeric like above
new = pd.DataFrame([[60, 'male', 27, 130, 200]], columns=['age', 'gender', 'bmi', 'blood_pressure',
'cholesterol'])
new['gender'] = le.transform(new['gender']) # Convert 'male' to numeric (1)

```

```
# Scale new input
newscale = scaler.transform(new)

# Predict probability
newcondition = model.predict_proba(newscale)[0][1]
print(f'Probability of developing the condition: {newcondition:.2f}')
```

Accuracy: 1.0

Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	1
1	1.00	1.00	1.00	2
accuracy			1.00	3
macro avg	1.00	1.00	1.00	3
weighted avg	1.00	1.00	1.00	3





