```
import numpy as np
import pandas as pd
from sklearn.datasets import make_classification
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, classification_report
```

```
# 1. Define feature and label names for context
feature_names = [
    'Age',
    'BMI',
    'Blood_Pressure',
    'Cholesterol',
    'Family_History_Score',
    'Physical_Activity_Level'
]
label_names = {0: 'Low Risk', 1: 'Medium Risk', 2: 'High Risk'}
```

```
# 2. DataFrame of viewing sample records
df = pd.DataFrame(X, columns=feature_names)
df['Risk_Level'] = pd.Series(y).map(label_names)
df.head()
```

	Age	BMI	Blood_Pressure	Cholesterol	Family_History_Score	Physical_Activity_Level	Risk_Level
0	-0.068404	-1.973482	0.199934	-1.516038	-1.055640	-0.053547	High Risk
1	2.329021	-0.461970	-0.126590	1.968747	-1.570531	-0.778238	Low Risk
2	1.285011	1.083324	0.035424	1.441720	-0.715682	-0.730726	High Risk
3	-0.937879	-0.365224	-0.312381	-0.094718	1.367096	0.913038	Low Risk
4	1.175840	-0.954824	-0.068427	0.664378	-1.004337	-0.308054	Low Risk

Next steps: Generate code with df | New interactive sheet

```
df.describe()
```

	Age	BMI	Blood_Pressure	Cholesterol	Family_History_Score	Physical_Activity_Level
count	800.000000	800.000000	800.000000	800.000000	800.000000	800.000000
mean	0.802431	-0.353972	-0.086168	0.698415	-0.495263	-0.183802
std	1.343957	0.795506	0.208841	1.013274	1.320258	0.705963
min	-3.954611	-1.973482	-0.830969	-2.406425	-4.032948	-2.758557
25%	-0.076672	-0.820784	-0.223032	0.060695	-1.430846	-0.661878
50%	0.756548	-0.574670	-0.094564	0.702712	-0.391677	-0.162842
75%	1.708798	-0.248114	0.043469	1.336944	0.416809	0.295248
max	4.675359	3.976145	0.745388	3.621082	4.388970	2.495277

```
# 3. Split the data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=21)
```

```
# 4. Train multiclass logistic regression
clf = LogisticRegression(multi_class='multinomial', solver='lbfgs', max_iter=200)
clf.fit(X_train, y_train)
```

```
/usr/local/lib/python3.12/dist-packages/sklearn/linear_model/_logistic.py:1247: FutureWarning: 'multi_class' was depr
  warnings.warn(
```

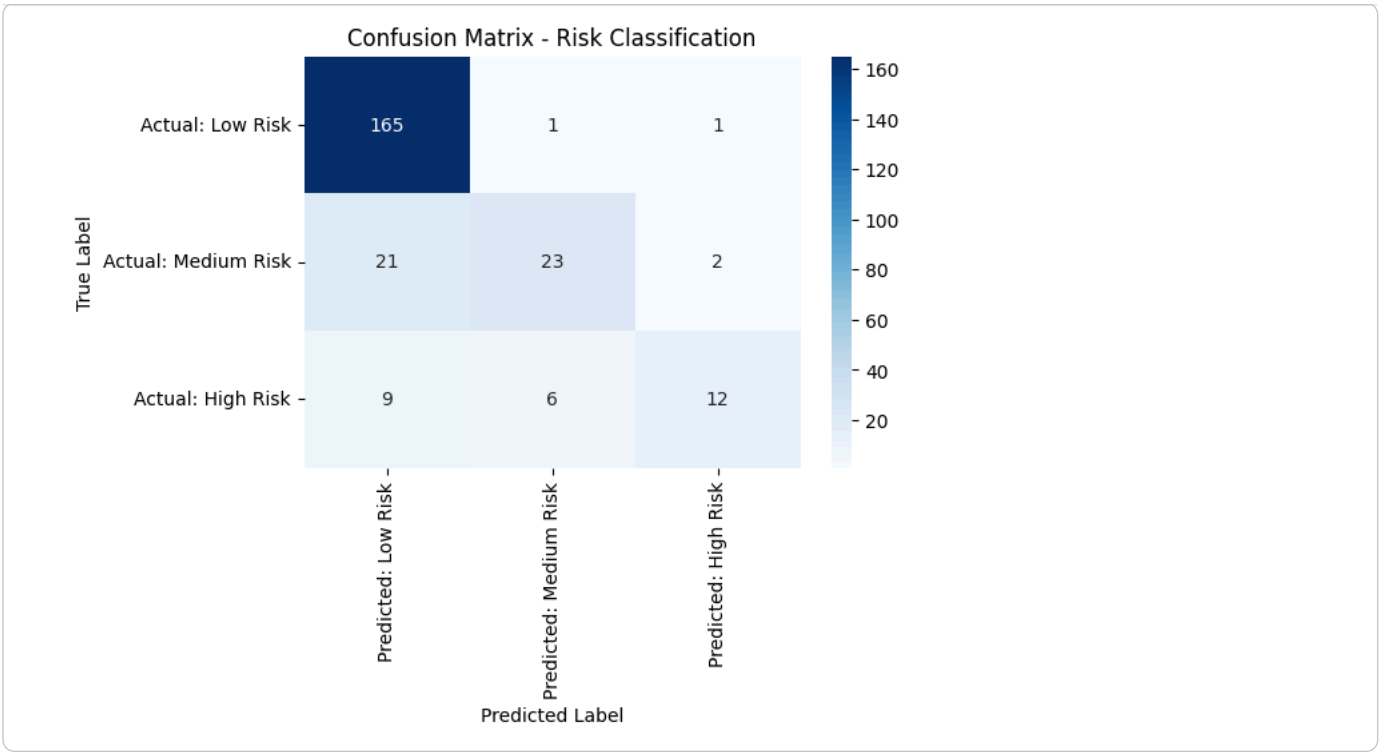
```
LogisticRegression
LogisticRegression(max_iter=200, multi_class='multinomial')
```

```
# 5. Make predictions and calculate metrics
y_pred = clf.predict(X_test)


# 6. Print accuracy and explicit metric values
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred, average=None, zero_division=0)
recall = recall_score(y_test, y_pred, average=None, zero_division=0)
f1 = f1_score(y_test, y_pred, average=None, zero_division=0)

# print(f'Overall Accuracy: {accuracy:.4f}')
# for i, cname in label_names.items():
#     print(f'Precision ({cname}): {precision[i]:.4f}, Recall: {recall[i]:.4f}, F1: {f1[i]:.4f}')


# Tabulate results using pandas DataFrame (for a pretty printed table)
summary_df = pd.DataFrame({
    'Class': [label_names[i] for i in range(3)],
    'Precision': precision,
    'Recall': recall,
    'F1-score': f1
})


print("Accuracy:", accuracy)
print(summary_df)
```

```
Accuracy: 0.8333333333333334
         Class  Precision    Recall  F1-score
0     Low Risk   0.846154  0.988024  0.911602
1  Medium Risk   0.766667  0.500000  0.605263
2    High Risk   0.800000  0.444444  0.571429
```

```
# 7. Confusion Matrix

from sklearn.metrics import confusion_matrix

# Generate confusion matrix
cm = confusion_matrix(y_test, y_pred)

# Optionally, display as a labeled DataFrame for clarity
cm_df = pd.DataFrame(
    cm,
    index=[f"Actual: {label_names[i]}" for i in range(3)],
    columns=[f"Predicted: {label_names[i]}" for i in range(3)]
)
```

```
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(6,4))
sns.heatmap(cm_df, annot=True, fmt='d', cmap='Blues')
plt.title('Confusion Matrix - Risk Classification')
plt.ylabel('True Label')
plt.xlabel('Predicted Label')
plt.show()
```


Confusion Matrix - Risk Classification
Actual: Low Risk
Actual: Medium Risk
Actual: High Risk
True Label
165
1
1
21
23
2
9
6
12
160
140
120
100
80
60
40
20
Predicted: Low Risk
Predicted: Medium Risk
Predicted: High Risk
Predicted Label