```
# Import necessary libraries

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.datasets import load_diabetes

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import StandardScaler

from sklearn.linear_model import LogisticRegression

from sklearn.metrics import accuracy_score, classification_report, confusion_matrix, roc_curve,
```

Double-click (or enter) to edit

## Read and explore the data

```
# Load the diabetes dataset

diabetes = load_diabetes()

X, y = diabetes.data, diabetes.target

# Convert the target variable to binary (1 for diabetes, 0 for no diabetes)

y_binary = (y > np.median(y)).astype(int)
```

## Splitting the dataset: Test and train models

```
# Split the data into training and testing sets

X_train, X_test, y_train, y_test = train_test_split(

X, y_binary, test_size=0.2, random_state=42)
```

## Feature Scaling

```
# Standardize features

scaler = StandardScaler()

X_train = scaler.fit_transform(X_train)
X_train = scaler.fit_transform(X_train)

X_test = scaler.transform(X_test)
```

## Train the model

```
# Train the Logistic Regression model

model = LogisticRegression()
model.fit(X_train, y_train)
```

```
▾ LogisticRegression
LogisticRegression()
```

## Evaluation matrix

```
# Evaluate the model

y_pred = model.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)

print("Accuracy: {:.2f}%".format(accuracy * 100))
```

```
Accuracy: 73.03%
```

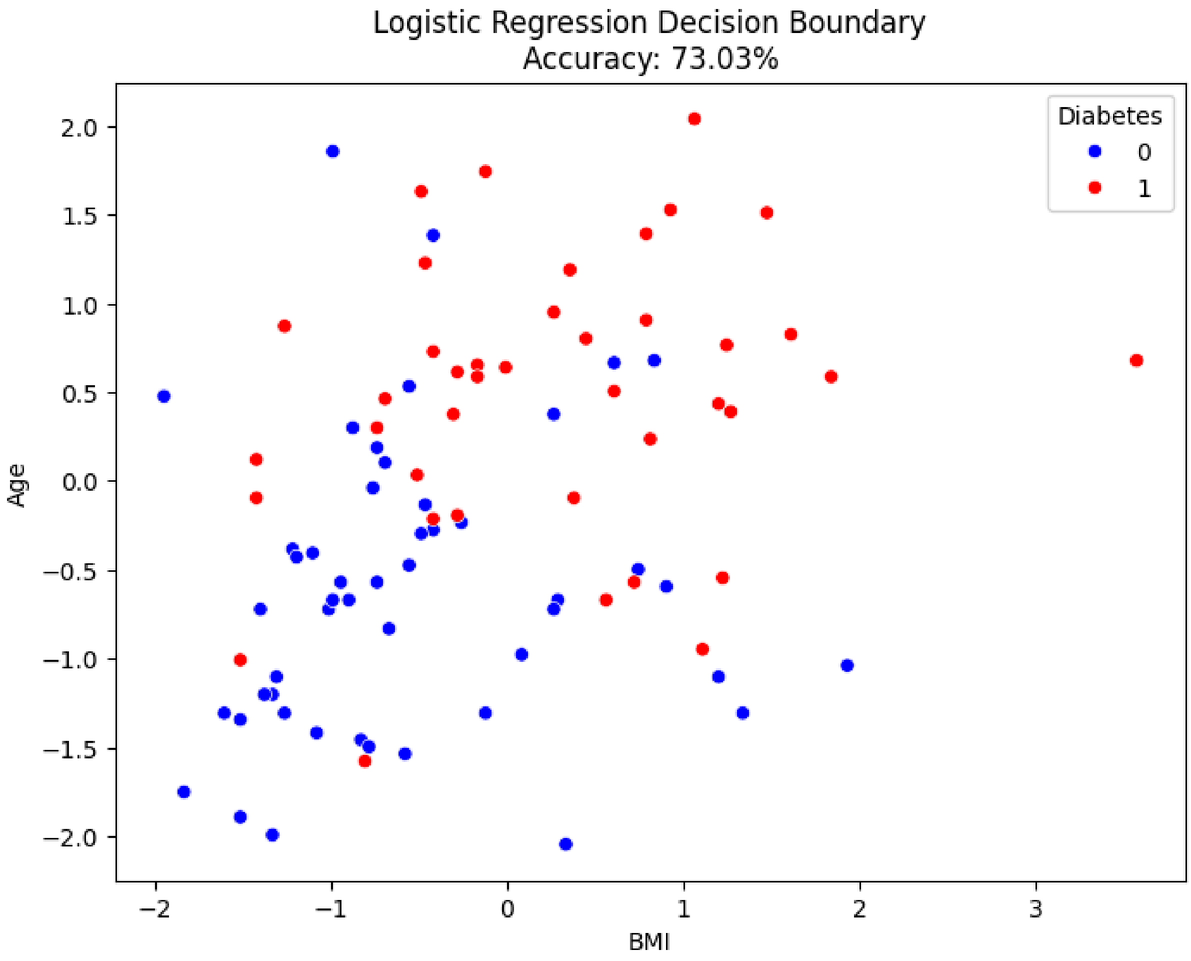
## Confusion matrix

```
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))

print("\nClassification Report:\n", classification_report(y_test, y_pred))
```

```
Confusion Matrix:
 [[36 13]
 [11 29]]

Classification Report:
               precision    recall  f1-score   support

           0       0.77      0.73      0.75        49
           1       0.69      0.72      0.71        40

    accuracy                           0.73        89
   macro avg       0.73      0.73      0.73        89
weighted avg       0.73      0.73      0.73        89
```

## Visualization the performance of our model

```
# Visualize the decision boundary with accuracy information

plt.figure(figsize=(8, 6))

sns.scatterplot(x=X_test[:, 2], y=X_test[:, 8], hue=y_test, palette={

                0: 'blue', 1: 'red'}, marker='o')

plt.xlabel("BMI")

plt.ylabel("Age")

plt.title("Logistic Regression Decision Boundary\nAccuracy: {:.2f}%".format(

    accuracy * 100))

plt.legend(title="Diabetes", loc="upper right")
plt.show()
```


Logistic Regression Decision Boundary
Accuracy: 73.03%
Diabetes
● 0
● 1
Age
BMI