In [1]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import confusion_matrix, accuracy_score, precision_score, r
```

In [2]:
```
# --- Load the dataset ---
df = pd.read_csv("diabetes.csv")  # Make sure 'diabetes.csv' is in the same fold
```

In [3]:
```
# --- Features and Target ---
X = df.drop(columns=['Outcome']).values  # All columns except target
y = df['Outcome'].values  # Target column
```

In [4]:
```
# --- Feature Scaling ---
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

In [5]:
```
# --- Train-Test Split ---
X_train, X_test, y_train, y_test = train_test_split(
    X_scaled, y, test_size=0.3, random_state=42, stratify=y
)
```

In [6]:
```
# --- K-Nearest Neighbors ---
knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train, y_train)
y_pred = knn.predict(X_test)
```

In [7]:
```
# --- Evaluation Metrics ---
acc = accuracy_score(y_test, y_pred)
err_rate = 1 - acc
prec = precision_score(y_test, y_pred, zero_division=0)
rec = recall_score(y_test, y_pred, zero_division=0)
cm = confusion_matrix(y_test, y_pred)
```

In [8]:
```
# --- Clean Output ---
print("--- K-Nearest Neighbors Performance on Diabetes Dataset ---")
print(f"Accuracy: {acc:.4f}")
print(f"Error Rate: {err_rate:.4f}")
print(f"Precision: {prec:.4f}")
print(f"Recall (Sensitivity): {rec:.4f}\n")
print("Confusion Matrix:")
print(cm)
```

```
--- K-Nearest Neighbors Performance on Diabetes Dataset ---
Accuracy: 0.7143
Error Rate: 0.2857
Precision: 0.6154
Recall (Sensitivity): 0.4938

Confusion Matrix:
[[125  25]
 [ 41  40]]
```

In [ ]: