In [1]:
```
import pandas as pd
import numpy as np
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_sco
```

In [2]:
```
df = pd.read_csv("diabetes.csv")
```

In [3]:
```
df = df.apply(pd.to_numeric, errors='coerce')
df.fillna(df.mean(),inplace=True)
```

In [4]:
```
X = df.drop('Outcome', axis=1)
y = df['Outcome'].astype(int)
```

In [5]:
```
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2, random_st
```

In [6]:
```
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

In [7]:
```
k=5
knn=KNeighborsClassifier(n_neighbors=k)
knn.fit(X_train, y_train)
```

Out[7]:
```
▾ KNeighborsClassifier

KNeighborsClassifier()
```

In [8]:
```
y_pred = knn.predict(X_test)
```

In [9]:
```
cm = confusion_matrix(y_test,y_pred)
print("Confusion Matrix:\n",cm)
```

```
Confusion Matrix:
 [[79 20]
 [27 28]]
```

In [10]:
```
accuracy = accuracy_score(y_test, y_pred)
error_rate = 1-accuracy
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
```

In [11]:

```
print(f'Accuracy:{accuracy*100:.2f}%')
print(f'Error Rate:{error_rate*100:.2f}%')
print(f'Precision:{precision:.2f}')
print(f'Recall:{recall:.2f}')
```

```
Accuracy:69.48%
Error Rate:30.52%
Precision:0.58
Recall:0.51
```

In [ ]: