34. Scenario: Suppose you are working as a data scientist for a medical research organization.

Your team has collected data on patients with a certain medical condition and their treatment

outcomes. The dataset includes various features such as age, gender, blood pressure, cholesterol

levels, and whether the patient responded positively ("Good") or negatively ("Bad") to the

treatment. The organization wants to use this model to identify potential candidates who are likely

to respond positively to the treatment and improve their medical approach.

Question: Your task is to build a classification model using the KNN algorithm to predict the

treatment outcome ("Good" or "Bad") for new patients based on their features. Evaluate the model's

performance using accuracy, precision, recall, and F1-score.Make predictions on the test set and

display the results.

Code:

**import** pandas **as** pd

**from** sklearn.model\_selection **import** train\_test\_split

**from** sklearn.preprocessing **import** StandardScaler, LabelEncoder

**from** sklearn.neighbors **import** KNeighborsClassifier

**from** sklearn.metrics **import** accuracy\_score, precision\_score, recall\_score, f1\_score, classification\_report

*# Load the data*

df **=** pd.read\_csv(r"C:\Users\vara prasad\Downloads\patient\_data.csv")

*# Encode categorical variables*

le\_gender **=** LabelEncoder()

df['Gender'] **=** le\_gender.fit\_transform(df['Gender']) *# Male=1, Female=0*

*# Features and target*

X **=** df[['Age', 'Gender', 'BloodPressure', 'Cholesterol']]

y **=** df['Outcome']

y **=** LabelEncoder().fit\_transform(y) *# Good=1, Bad=0*

*# Split the data*

X\_train, X\_test, y\_train, y\_test **=** train\_test\_split(X, y, test\_size**=**0.3, random\_state**=**42)

*# Feature scaling*

scaler **=** StandardScaler()

X\_train\_scaled **=** scaler.fit\_transform(X\_train)

X\_test\_scaled **=** scaler.transform(X\_test)

*# Train the model*

knn **=** KNeighborsClassifier(n\_neighbors**=**3)

knn.fit(X\_train\_scaled, y\_train)

*# Predict and evaluate*

y\_pred **=** knn.predict(X\_test\_scaled)

*# Metrics*

print("Accuracy:", accuracy\_score(y\_test, y\_pred))

print("Precision:", precision\_score(y\_test, y\_pred))

print("Recall:", recall\_score(y\_test, y\_pred))

print("F1 Score:", f1\_score(y\_test, y\_pred))

print("\nClassification Report:\n", classification\_report(y\_test, y\_pred))

*# Optional: Show predictions*

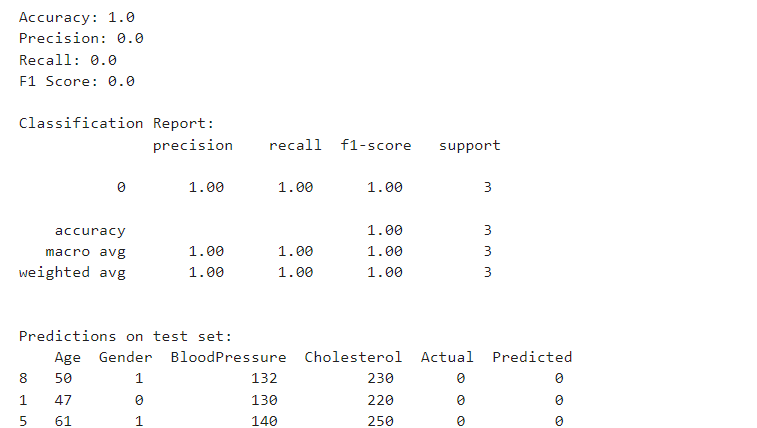
results **=** X\_test.copy()

results['Actual'] **=** y\_test

results['Predicted'] **=** y\_pred

print("\nPredictions on test set:\n", results)

Output:



Dataset:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Age | Gender | BloodPressure | Cholesterol | Outcome |
| 25 | Male | 120 | 180 | Good |
| 47 | Female | 130 | 220 | Bad |
| 52 | Female | 135 | 210 | Bad |
| 36 | Male | 128 | 190 | Good |
| 29 | Female | 118 | 170 | Good |
| 61 | Male | 140 | 250 | Bad |
| 45 | Male | 138 | 240 | Bad |
| 38 | Female | 125 | 200 | Good |
| 50 | Male | 132 | 230 | Bad |
| 60 | Female | 145 | 260 | Bad |