ML_4

November 11, 2025

```
[1]: import pandas as pd
     import numpy as np
     from sklearn.model_selection import train_test_split
     from sklearn.metrics import accuracy_score, confusion_matrix, precision_score,␣
      ↪recall_score,classification_report
     from sklearn.preprocessing import StandardScaler
     from sklearn.neighbors import KNeighborsClassifier
```

```
[3]: df = pd.read_csv(r"C:
      ↪\Users\suraj\OneDrive\Desktop\LP3-master\ML\datasets\diabetes.csv")
```

```
[5]: df.head()
```

```
[5]:    Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  \
     0            6      148             72             35        0  33.6
     1            1       85             66             29        0  26.6
     2            8      183             64              0        0  23.3
     3            1       89             66             23       94  28.1
     4            0      137             40             35      168  43.1

        Pedigree  Age  Outcome
     0     0.627   50        1
     1     0.351   31        0
     2     0.672   32        1
     3     0.167   21        0
     4     2.288   33        1
```

```
[7]: df.shape
```

```
[7]: (768, 9)
```

```
[9]: df.isna().sum()
```

```
[9]: Pregnancies      0
     Glucose          0
     BloodPressure    0
     SkinThickness    0
     Insulin          0
```

```
BMI                 0
Pedigree            0
Age                 0
Outcome             0
dtype: int64
```

[11]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Pregnancies    768 non-null    int64
 1   Glucose        768 non-null    int64
 2   BloodPressure  768 non-null    int64
 3   SkinThickness  768 non-null    int64
 4   Insulin        768 non-null    int64
 5   BMI            768 non-null    float64
 6   Pedigree       768 non-null    float64
 7   Age            768 non-null    int64
 8   Outcome        768 non-null    int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

[13]:
```
X = df.drop('Outcome', axis=1)
y = df['Outcome']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.
 ↪2,random_state=42)
```

[15]:
```
scaler = StandardScaler()

X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train_scaled, y_train)

y_pred = knn.predict(X_test_scaled)
```

[17]:
```
cm = confusion_matrix(y_test, y_pred)
print("confusion matrix:\n",cm)
```

```
confusion matrix:
 [[79 20]
 [27 28]]
```

[24]:
```
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)

print(f"Accuracy :{accuracy:.4f}")
print(f"Error rate : {error_rate: .4f}" )
print(f"Precision : {precision: .4f}")
print(f"Recall : {recall: .4f}")
```

```
Accuracy :0.6948
Error rate :  0.3052
Precision :  0.5833
Recall :  0.5091
```

[26]:
```
print("KNN Classification Report:\n", classification_report(y_test, y_pred,␣
 ↪zero_division=0))
```

```
KNN Classification Report:
               precision    recall  f1-score   support

           0       0.75      0.80      0.77        99
           1       0.58      0.51      0.54        55

    accuracy                           0.69       154
   macro avg       0.66      0.65      0.66       154
weighted avg       0.69      0.69      0.69       154
```

[ ]:
```

```