

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.preprocessing import StandardScaler
import warnings
warnings.filterwarnings('ignore')
```

```
#df = pd.read_csv("corrections.csv")
df = pd.read_excel("corrections1.xlsx")
```

```
df.head()
```



	Horodateur	nom prénom	numero du dossier	date de l'intervention	Age lors de l'intervention	sexe	IMC (Kg/ m2)	SC (m2)
0	2025-04-26 13:12:53.526	Tadj ATTALAH	191/2024	2024-08-07	65	1	28	1.93
1	2025-04-26 13:32:03.839	KEBAILI Ahmed	194/2024	2024-07-14	64	1	23.9	1.82
2	2025-04-26 13:45:11.675	SALHI Sidahmed	176/2024	2024-07-02	46	1	31	2.22
3	2025-04-26 13:57:31.308	BREIDJ Nacer	85/2024	2024-05-12	60	1	24.1	1.69
4	2025-04-26 14:14:36.602	ZEGHIM Elhoussine	112/2024	2024-04-28	67	1	27.5	1.82

5 rows × 44 columns

```
df.info()
```



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 39 entries, 0 to 38
Data columns (total 44 columns):
#   Column                                     Non-Null Count  Dtype
---  -
0   Horodateur                               39 non-null     datetime64
1   nom prénom                              39 non-null     object
2   numero du dossier                       39 non-null     object
3   date de l'intervention                  39 non-null     datetime64
4   Age lors de l'intervention              39 non-null     int64
```

```

5  sexe                                     39 non-null    int64
6  IMC (Kg/m2)                             37 non-null    object
7  SC (m2)                                 38 non-null    float64
8  OBESITE                                 38 non-null    float64
9  Comorbidités                             39 non-null    int64
10 Traitements médicamenteux pré-opératoires : 33 non-null    object
11 Surface valvulaire aortique : cm²         38 non-null    float64
12 Vitesse max : m/s                         39 non-null    float64
13 Gradient moyen transvalvulaire : mmHg     39 non-null    int64
14 Fraction d'éjection VG : %                38 non-null    float64
15 Volume télédiastolique VG : ml/m2         38 non-null    float64
16 Volume télésystolique VG : ml/m2          38 non-null    float64
17 Volume OG : ml/m2                         35 non-null    float64
18 PAPS : mmHg                               36 non-null    float64
19 Autres anomalies associées :               26 non-null    object
20 Type de chirurgie :                       38 non-null    object
21 Type de prothèse implantée                 39 non-null    object
22 Marque et Numéro de prothèse               39 non-null    object
23 Temps de circulation extracorporelle : min 34 non-null    float64
24 Temps de clampage aortique : min           26 non-null    float64
25 Vitesse max prothese : m/s                 33 non-null    float64
26 Gradient moyen protheses : mmHg           37 non-null    float64
27 Surface de la prothèse : cm²               33 non-null    float64
28 Surface indexée : cm²/m²                  33 non-null    float64
29 Fraction d'éjection VG post op: %          31 non-null    float64
30 PAPS post-opératoire : mmHg                25 non-null    float64
31 Complications post-opératoires précoces (< 30 jours) 39 non-null    int64
32 Durée de séjour en reanimation             39 non-null    int64
33 Durée de séjour en unite post opératoire   39 non-null    int64
34 Durée de séjour total                      39 non-null    int64
35 Décès en post-opératoire                   39 non-null    int64
36 masse VG g/m2                             37 non-null    float64
37 Surface valvulaire aortique indexée : cm²/m2 38 non-null    float64
38 ANATOMIE DE LA VALVE ECHO                  33 non-null    object
39 ANATOMIE DE LA VALVE CHIRURGIE             20 non-null    object
40 STRAIN VG GLS : %                          21 non-null    object
41 ATCD CHIRURGICAUX CARDIAQUES               2 non-null     object
42 CHIRURGIE CARDIAQUE PRECEDENTE:            1 non-null     object
43 DIAMETRE SOUS AO OU ANNEAU AO             22 non-null    float64
dtypes: datetime64[ns](2), float64(20), int64(9), object(13)
memory usage: 13.5+ KB

```

```

X = df[[ "masse VG g/m2",
        "Temps de circulation extracorporelle : min",
        "Gradient moyen protheses : mmHg","Surface de la prothèse : cm²"]]

```

```

y = df["Complications post-opératoires précoces (< 30 jours)"]

```

```

# Nettoyage (si valeurs manquantes)
X = X.dropna()
y = y.loc[X.index]

```

```

# Diviser en training/test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

y_test.shape

(9,)

# Standardisation
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# KNN classifier
knn = KNeighborsClassifier(n_neighbors=4)
knn.fit(X_train_scaled, y_train)

# Prédiction
y_pred = knn.predict(X_test_scaled)

# Évaluation
from sklearn.metrics import confusion_matrix, classification_report
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))

```

```

[[8 0]
 [1 0]]

```

	precision	recall	f1-score	support
0	0.89	1.00	0.94	8
1	0.00	0.00	0.00	1
accuracy			0.89	9
macro avg	0.44	0.50	0.47	9
weighted avg	0.79	0.89	0.84	9

Commencez à coder ou à [générer](#) avec l'IA.

Commencez à coder ou à [générer](#) avec l'IA.

