Good for small things:

K-NN : {K - Nearest Neighbors}

K → No. of Nearest Neighbors needed to classify something

to choose k : $k = \sqrt{n}$ n: total points
and if k is even
k+=1

Euclidean distance: $\sqrt{(x-a)^2 + (y-b)^2}$

Weight	Height	Class	Distance
51	162	Under	6.7
62	182	Normal	13
69	176	Normal	13.4
64	173	Normal	7.6
65	172	Normal	8.2
56	174	Under	4.1
56	169	Normal	1.4
57	173	Normal	3
55	170	Normal	2

k=3

for(57, 170)

Date :   Mo Tu We Th Fr Sa Su

for many attributes :

Pregnencies Glucose BP Skin Insulin

BMI Diabetes test Age Outcome:

~~where~~ Steps:

→ Wherever zero replace with NULL

→ Mean of column (skip NULL)

→ Replace NULL with respective Mean.

→ Splite dataset into training and testing part.

→ X → all inputs Y → output

→ test_size : % of data for testing

→ Scale data (standardize)

→ Scale X-test & X_train

→ StandardScaler().fit_transform (X_train)

→ Use KNN:

classifier = KNeighborsClassifier

k ↓ outputs.

(N_neighbors = 11, p = 2,

metric = 'euclidean')

↑

distance.

classifier.fit (X_train, y_train)

→ Evaluate Model (confusion Matrix)