Checking for normality in a feature P>0.05 feature seems normal within this class P< 0.05 Feature distribution likely not normal

```
from scipy.stats import shapiro

for cls in df['diabetes'].unique():
    data = df[df['diabetes'] == cls]['sbp'].dropna()
    stat, p = shapiro(data)
    print(f"Class {cls}: Shapiro-Wilk test → W={stat:.3f}, p={p:.3f}")
```

1. LDA — Linear Discriminant Analysis Type

Linear classifier

Parametric (assumes a specific data distribution)

 Key idea

LDA assumes:

Each class's features are normally distributed,

All classes share the same covariance matrix (same "shape" of distribution, just different centers).

It finds a linear boundary (a straight line or plane) that best separates classes.

 Decision boundary

Linear → looks like a straight line in 2D, or a hyperplane in higher dimensions.

 Analogy

It's like drawing a straight line that separates red and blue points, assuming both groups are "blobs" of normally distributed data.

 Pros

Fast, interpretable

Works well if data roughly meets the normality & equal covariance assumptions

Good for small datasets

 Cons

Not good if boundaries are curved or nonlinear

Assumptions may fail in real-world data

 2. QDA — Quadratic Discriminant Analysis Type

Nonlinear classifier

Parametric (also assumes normal distributions)

 Key idea

Same as LDA, but allows each class to have its own covariance matrix.

That means it can fit curved (quadratic) decision boundaries — more flexible.

 Decision boundary

Curved (parabolic, elliptical, etc.)

 Analogy

If your data looks like this:

Red Blue ○ ● ○ ● ○● ○ ● ○ ●

LDA draws a straight line; QDA can draw a curved line that better separates the two.

 Pros

Captures nonlinear relationships

More flexible than LDA

 Cons

Needs more data (more parameters to estimate)

Can overfit if dataset is small

 3. SVM — Support Vector Machine Type

Linear or nonlinear classifier

Non-parametric (no distribution assumptions)

 Key idea

Finds the maximum-margin hyperplane — the boundary that separates classes with the largest gap between them.

If data aren't linearly separable, SVM can use a kernel trick (like polynomial or RBF) to project data into a higher-dimensional space where a linear separator exists.

 Decision boundary

Can be linear or nonlinear, depending on kernel.

 Analogy

Imagine two groups of points — SVM finds the "widest street" that separates them and places the decision boundary right in the middle.

 Pros

Works well on complex, nonlinear data

Doesn’t need normality assumptions

Often high accuracy

 Cons

Harder to interpret

Sensitive to parameter tuning (especially kernel and C)

Slower on very large datasets

 4. k-NN — k-Nearest Neighbors Type

Non-parametric, instance-based classifier

 Key idea

When predicting a class for a new point:

Find the k closest training points (neighbors) in feature space.

Assign the majority class among those neighbors.

So it makes predictions based purely on proximity.

 Decision boundary

Highly nonlinear, follows the shape of the data.

 Analogy

If you move into a new neighborhood, your class (e.g., “diabetic or not”) is predicted based on the majority of your nearest “neighbors”.

 Pros

Simple and intuitive

No training phase (just stores data)

Works well if classes are locally well-separated

 Cons

Slow for large datasets (must compare to every training sample)

Sensitive to scaling (need to normalize features)

Sensitive to noisy or irrelevant features

Quick Comparison Summary Algorithm Linear? Assumptions Boundary Type Strengths Weaknesses

LDA Yes Normal distribution, equal covariance Linear Fast, interpretable Poor with nonlinear data

QDA □ No Normal distribution Quadratic (curved) Flexible Needs more data

SVM □/□ (depends on kernel) None Linear or nonlinear Robust, powerful Needs tuning

k-NN □ No None Nonlinear, jagged Simple, effective locally Slow, sensitive to noise

. . . □ When to use what

Situation Best Choice

Features roughly normal, linear separation LDA

Features normal but curved separation QDA

Complex nonlinear patterns SVM (with kernel)

Small dataset, few assumptions k-NN

You want interpretability LDA or logistic regression

```
from Extra_python import MethodRecommender
import pandas as pd

df = pd.read_csv("collegiate_athlete_injury_dataset.csv")

print(df.head())

print(df.isna().sum()[df.isna().sum() > 0])

recommend = MethodRecommender(df, "Gender")
recommend.run()
```

```
  Athlete_ID  Age  Gender  Height_cm  Weight_kg Position
Training_Intensity  \
0       A001   24  Female        195         99   Center
2
1       A002   21    Male        192         65  Forward
8
2       A003   22    Male        163         83    Guard
8
3       A004   24  Female        192         90    Guard
1
4       A005   20  Female        173         79   Center
3

   Training_Hours_Per_Week  Recovery_Days_Per_Week
Match_Count_Per_Week  \
0                       13                       2
3
1                       14                       1
3
2                        8                       2
1
```

```
3                         13                          1
1
4                          9                          1
2

   Rest_Between_Events_Days  Fatigue_Score  Performance_Score  \
0                         1              1                 99
1                         1              4                 55
2                         3              6                 58
3                         1              7                 82
4                         1              2                 90

   Team_Contribution_Score  Load_Balance_Score  ACL_Risk_Score  \
0                       58                 100               4
1                       63                  83              73
2                       62                 100              62
3                       74                  78              51
4                       51                  83              49

   Injury_Indicator
0                 0
1                 0
2                 0
3                 0
4                 0
Series([], dtype: int64)

--- Answer the following questions ---

Do you believe the relationship is linear?
1. yes
2. no

Choose a number:  1


Do you need high interpretability?
1. yes
2. no

Choose a number:  2


-------------------------------------
 Recommended Method: LDA or SVM
-------------------------------------
```