

Problem Definition: Sales Stability & Risk Assessment System

1. Scenario

A multinational retail corporation tracks its inventory performance across different geographical regions. The data is stored in a 3D NumPy array of shape (6, 5, 2), where:

- Axis 0: Represents 6 Products.
- Axis 1: Represents 5 Regions.
- Axis 2: Represents 2 Metrics (Index 0: Total Units Sold, Index 1: Units Returned).

You are required to build an analytics engine using Single Inheritance to evaluate product stability and identify operational risks.

```
sales = np.array([
    [[120,10], [140,12], [130,11], [150,13], [135,10]],
    [[80,15], [90,18], [85,16], [88,17], [92,19]],
    [[200,8], [210,9], [205,10], [215,8], [220,7]],
    [[60,20], [65,22], [70,21], [68,23], [72,24]],
    [[150,12], [155,13], [148,14], [160,15], [158,12]],
    [[230,6], [240,7], [235,6], [245,8], [250,7]]
])
```

2. Class Structure

A. Base Class: BaseSalesData

- Purpose: To initialize the core data and store basic array dimensions.
- Attributes: sales (3D Array), products (Integer), and regions (Integer).

B. Derived Class: SalesStabilityAnalysis

- Inheritance: Inherits from BaseSalesData.
- Purpose: Implements logic for net sales, stability indexing, and priority planning.

3. SalesStabilityAnalysis methods: Functional Requirements & Return Types

Method Name	Objective	Logic / Criteria	Output Return Data Type
net_units()	Calculate actual sales.	Subtract Returns from Units Sold .	NumPy Array (2D)
risk_products()	Identify high-return items.	Returns > 15 in any region.	List

Method Name	Objective	Logic / Criteria	Output Return Data Type
<code>regional_reliability()</code>	Measure region performance.	Average net units per region.	List
<code>most_stable_product()</code>	Find consistent performers.	Highest minimum net units.	Integer
<code>normalize_returns()</code>	Data Cleaning.	Cap returns between 14-16 at 14 .	NumPy Array (2D)
<code>stability_index()</code>	Performance Scoring.	Divide the average net units by 200 and then multiply the result by 100	List
<code>forecast_readiness()</code>	Categorical Status.	High, Moderate, Low, or Unstable. Label as High (>75), Moderate (>60), Low (>50), or Unstable (<50).	List
<code>planning_priority()</code>	Strategy Ranking.	Product IDs sorted by Stability.	List

Note: For proper execution and correct output generation, it is mandatory to include the following code in your program.

```
sa = SalesStabilityAnalysis(sales)
```

```
print("Net Units:\n", sa.net_units())
print("Risk Products:", sa.risk_products())
print("Regional Reliability:", sa.regional_reliability())
print("Most Stable Product ID:", sa.most_stable_product())
print("After Return Normalization:\n", sa.normalize_returns())
print("Stability Index:", sa.stability_index())
print("Forecast Readiness:", sa.forecast_readiness())
print("Planning Priority:", sa.planning_priority())
```