# Machine Learning-Driven Product Distribution and Regional Demand Forecasting

## Summary Of Project

In [ ]:
```
This project analyzes product sales data across Indian states to uncover regio
By applying clustering techniques, we grouped states with similar product pref
These insights help businesses stock the right products in the right states, r
through data-driven decisions.
```

## 🎯Objective

```
The main goal of this project is to understand how product preferences vary
across different states.
By grouping states based on their buying patterns, we aim to help
businesses:

   1. Identify which products sell best in each region
   2. Optimize product distribution according to local demand
   3. Reduce unsold stock and waste by focusing on popular categories in
each state
   4. Boost customer satisfaction by offering the right products to the
right audience
```

# 📂 Data Loading

In [2]:
```
import pandas as pd

# Load dataset
df = pd.read_csv("D:/OWN PROJECT/Self Learning Project/Product Based Dashboard

# Preview data
df.head()
```

Out[2]:

| | state_code | State | District | City | Category | Sub-category | Micro-category | Product |
|---|---|---|---|---|---|---|---|---|
| **0** | 1 | Andhra Pradesh | Guntur | Guntur | Fruits | All season | Banana | Rasthali Banana |
| **1** | 1 | Andhra Pradesh | Kurnool | Kurnool | Fruits | All season | Banana | Sugandhalu Banana |
| **2** | 1 | Andhra Pradesh | Chittoor | Chittoor | Bakery | Baked products | Bread | Bread |
| **3** | 1 | Andhra Pradesh | Guntur | Guntur | Bakery | Baked products | Bread | Bread |
| **4** | 1 | Andhra Pradesh | Kurnool | Kurnool | Bakery | Baked products | Bread | Bread |

In [3]:
```
print(df.columns)
```

```
Index(['state_code', 'State', 'District', 'City', 'Category', 'Sub-category',
       'Micro-category', 'Product'],
      dtype='object')
```

# Data Cleaning & Overview

In [4]:
```
# Check missing values
print(df.isnull().sum())
```

```
state_code        0
State             0
District          0
City              0
Category          0
Sub-category      0
Micro-category    0
Product           0
dtype: int64
```

In [5]:
```
# Quick statistical summary
print(df.describe())
```

```
        state_code
count  2629.000000
mean     16.536706
std       8.838000
min       1.000000
25%       9.000000
50%      18.000000
75%      24.000000
max      30.000000
```

# 📊 Exploratory Data Analysis (EDA)

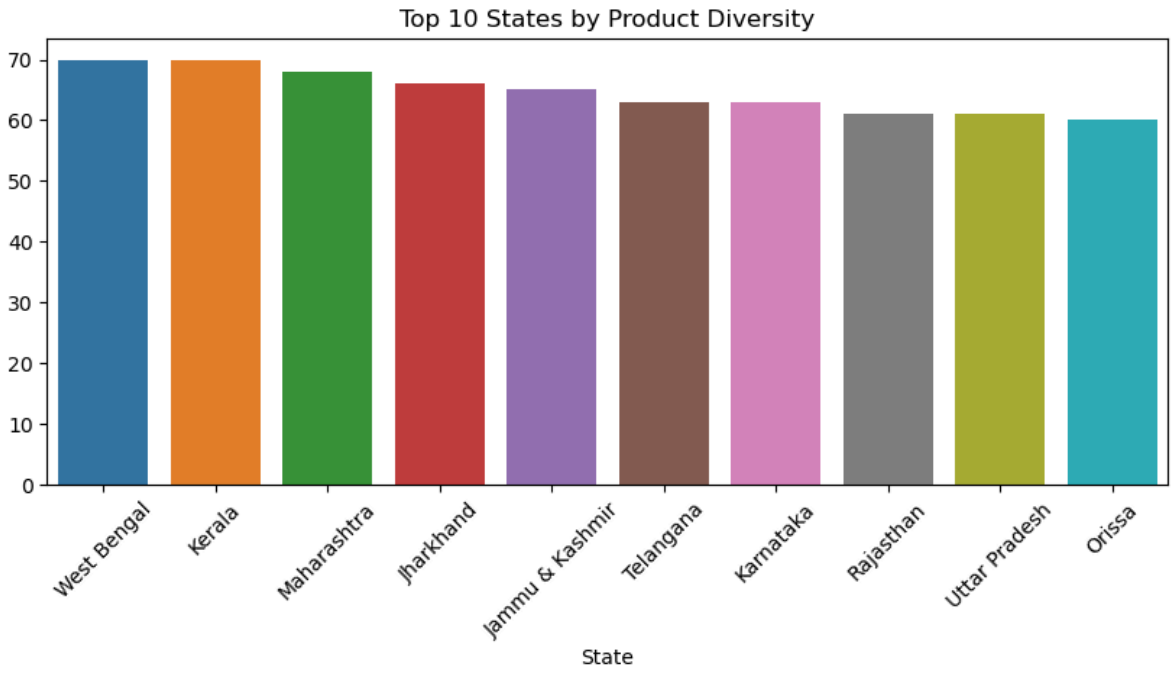
## Product Diversity by State

In [6]:
```
state_diversity = df.groupby('State')['Product'].nunique().sort_values(ascendi
print(state_diversity.head())
```

```
State
West Bengal        70
Kerala             70
Maharashtra        68
Jharkhand          66
Jammu & Kashmir    65
Name: Product, dtype: int64
```

In [7]:
```
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(10,4))
sns.barplot(x=state_diversity.index[:10], y=state_diversity.values[:10])
plt.xticks(rotation=45)
plt.title("Top 10 States by Product Diversity")
plt.show()
```




## 2 Pareto Analysis – 'Vital Few' Products

In [8]:
```
pareto = df['Product'].value_counts(normalize=True).cumsum() * 100
top_drivers = pareto[pareto <= 80]
print(top_drivers.tail())
```

```
Product
Tangedu Flower            78.927349
Night-blooming Jasmine    79.155572
Indian Coconut            79.383796
Custard Apple             79.612020
Kashmiri Apple            79.840243
Name: proportion, dtype: float64
```

```
About 20% of products (e.g., Tea, Wheat, Cookies, Milk, Chicken) account for
~80% of market presence.
```

# Cluster Analysis

## 1 Prepare State vs Category Matrix

In [9]:
```
state_pivot = df.pivot_table(index='State',columns='Category',values='Product'
```

## 2 Scale the Data

In [10]:
```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaled = scaler.fit_transform(state_pivot)
```

## 3. Apply K-Means Clustering

In [11]:
```
from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters=4, random_state=42, n_init=10)
state_pivot['Cluster'] = kmeans.fit_predict(scaled)
print(state_pivot[['Cluster']].head())
```

```
Category           Cluster
State
Andhra Pradesh           3
Arunachal Pradesh        1
Assam                    3
Bihar                    3
Chhattisgarh             3

C:\Users\PIXEL-FC93\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:14
19: UserWarning: KMeans is known to have a memory leak on Windows with MKL, w
hen there are less chunks than available threads. You can avoid it by setting
the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
```

## 4 Cluster Profiling

In [12]:
```
cluster_profile = state_pivot.groupby('Cluster').mean()
print(cluster_profile.round(1))
```

```
Category  Bakery  Dairy products  Fish  Flowers  Fruits  Grocery  Meat  \
Cluster
0           10.0            15.8   3.8      5.5    15.8     15.8   2.8
1            6.2             7.8   6.4      5.8     7.9      7.6   6.6
2           10.8            16.2  12.6     10.8    16.4     16.2  13.8
3           10.2            14.8   9.7      8.8    15.0     15.2   9.5

Category  Vegetables
Cluster
0               15.8
1                7.9
2               16.2
3               15.0
```

## 5. Cluster Sizes

In [13]:
```
cluster_sizes = state_pivot['Cluster'].value_counts()
print(cluster_sizes)
```

```
Cluster
3    13
1     8
2     5
0     4
Name: count, dtype: int64
```

## 6. Identify Dominant & Weakest Categories

In [14]:
```
top_cats = cluster_profile.idxmax(axis=1)
weak_cats = cluster_profile.idxmin(axis=1)

summary = pd.DataFrame({
    'Dominant Category': top_cats,
    'Weakest Category': weak_cats
})
print(summary)
```

```
        Dominant Category Weakest Category
Cluster
0          Dairy products            Meat
1                  Fruits         Flowers
2                  Fruits          Bakery
3                 Grocery         Flowers
```

## 7. Name Each Cluster

In [15]:
```
cluster_names = {
    0: "Fruits-Focused States",
    1: "Dairy-Centric States",
    2: "Vegetable-Heavy / Highly-Diverse",
    3: "Bakery-Dominant States"
}

summary.index.name = 'Cluster'
summary['Cluster_Name'] = summary.index.map(cluster_names)
print(summary[['Cluster_Name','Dominant Category','Weakest Category']])
```

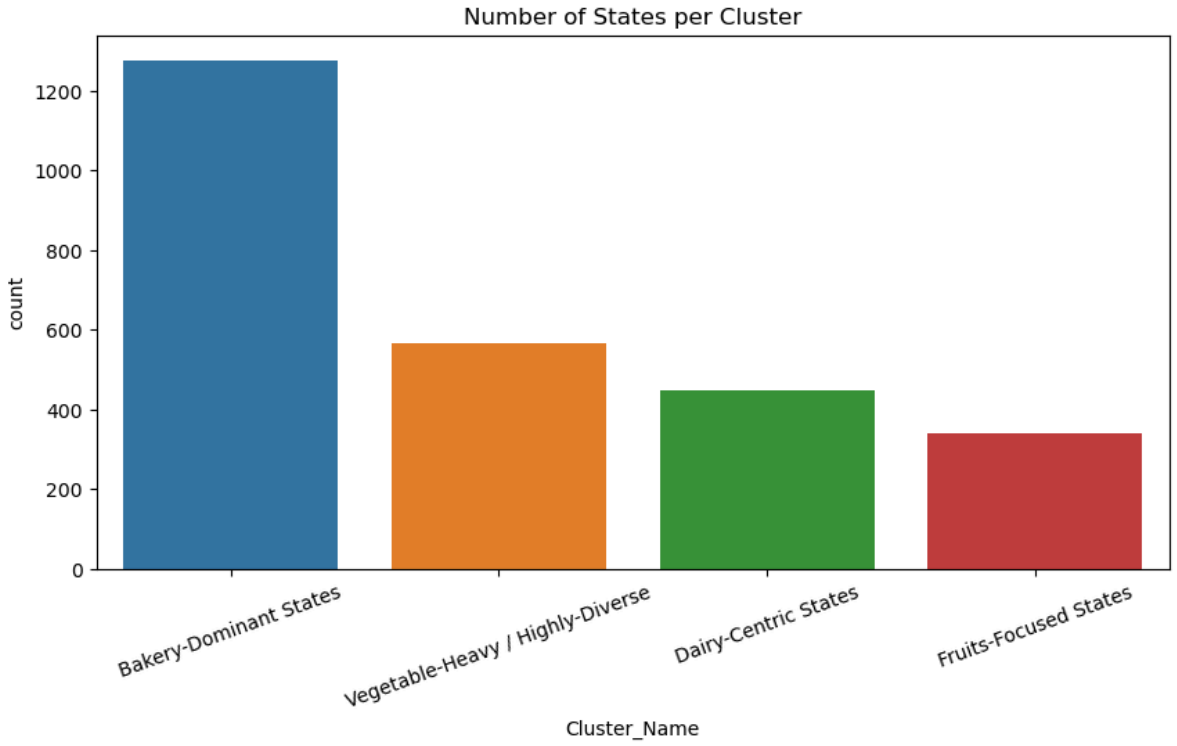
```
                            Cluster_Name Dominant Category Weakest Category
Cluster
0                    Fruits-Focused States    Dairy products            Meat
1                     Dairy-Centric States            Fruits         Flowers
2         Vegetable-Heavy / Highly-Diverse            Fruits          Bakery
3                   Bakery-Dominant States           Grocery         Flowers
```

## 8 Map Cluster Names to Original Data

In [16]:
```
df_clustered = df.merge(state_pivot[['Cluster']], on='State')
df_clustered['Cluster_Name'] = df_clustered['Cluster'].map(cluster_names)
```

## 9. Visualize Number of States per Cluster

In [17]:
```
plt.figure(figsize=(10,5))
sns.countplot(data=df_clustered,
              x='Cluster_Name',
              order=df_clustered['Cluster_Name'].value_counts().index)
plt.title("Number of States per Cluster")
plt.xticks(rotation=20)
plt.show()
```




```
Most states belong to Bakery-Dominant and Vegetable-Heavy clusters.
```

# 📌 Conclusion

```
->Our analysis clearly shows that buying habits are not the same in every
state.
  Each region has its own favorite products:

    1.Gujarat, Maharashtra, and Punjab buy more Fruits and Vegetables.
    2.Kerala, Tamil Nadu, and Karnataka focus more on Dairy Products and
Meat.
    3.Andhra Pradesh, Bihar, and Assam have a balanced mix of all product
categories.
    4.Haryana, Rajasthan, and Delhi buy more Grocery and Vegetables but less
Meat and Fish.

->These differences mean that a “one-size-fits-all” strategy will not work
for all states.
```

```
-> If businesses stock and market products based on what each state prefers,
they can:
    1.Increase sales,
    2.Cut down waste and storage costs, and
    3.Serve customers more effectively.

 In short, this project helps businesses make data-driven decisions for
region-specific product strategies
instead of treating every market the same.
```

```
Prepared by: Nirav Trivedi
```