NATA SUPERMARKET CASE

Business Analytics

TEAM 1
LAURA VICTORIA MIQUEL HERRERA
SAÚL JOSUÉ RUIZ GONZÁLEZ
RODRIGO RIVAS GONZÁLEZ
KARLA SOFÍA CANTÚ ZENDEJAS

INTRODUCTION

In January 2022, Vina Verago, Vice President of Technology at Nata Supermarkets, reviewed the company's performance for 2021.

The evaluation revealed that Nata was underperforming and **lagging behind** its competitors in both **overall growth** and **operational efficiency**. For example:

- Poor promotion targeting
- Inefficient product inventory management.

DATA

Nata, in its attempt to improve, provided us with a database containing various factors they consider important; therefore, our work will depend on properly using this database in order to implement methodologies to address their issues.

The core problem being the failure when identifying purchasing patterns and customer preferences, improvement of promotion effectiveness and the forecasting the product demand more accurately.

```
# Step 1: Drop rows where all values are NaN
df_clean = df.dropna(how='all')

# Step 2: Drop columns where NaN proportion > clean_rate
limit = clean_rate * len(df_clean)
df_clean = df_clean.loc[:, df_clean.isna().sum() <= limit]

# Step 3: Drop duplicate rows
df_clean = df_clean.drop_duplicates(keep='first')

# Step 4: Separate categorical and numerical columns
categorical_cols = df_clean.select_dtypes(include=['object', 'category']).columns
numerical_cols = df_clean.select_dtypes(include=['number']).columns

# Step 5: Fill missing values
# Categorical -> fill with mode
for col in categorical_cols:
    if df_clean[col].isna().sum() > 0:
        df_clean[col].fillna(df_clean[col].mode()[0], inplace=True)

# Numerical -> fill with mean
for col in numerical_cols:
    if df_clean[col].isna().sum() > 0:
        df_clean[col].fillna(df_clean[col].mean(), inplace=True)
```

STEP 1: DROP ROWS WITH ALL NANS

STEP 2: DROP COLUMNS WITH NAN PROPORTION > CLEAN_RATE

STEP 3: DROP DUPLICATE ROWS

STEP 4: FILL MISSING VALUES

STEP 5: CONVERT TO DATETIME

STEP 6: CHECK REMAINING MISSING VALUES

WHAT CAN WE DO TO SOLVE IT?

01

Clustering with K-Means

Groups similar data points together based on their characteristics, with this we can create different campaigns.

02

Forecasting method

It helps plan resources more efficiently between clusters and avoid shortages or excess inventory.

03

Inventory analysis

This method checks stock levels to balance supply and demand, avoiding waste and reducing storage costs. It improves cash flow and efficiency.

HOW TO KNOW THE VALUE OF K

01

METHOD SELECTION

- Applied Elbow Method and Silhouette Score to determine optimal K
- Tested K values from 2 to 10

02

RESULTS ANALYSIS

- Elbow Method: Shows gradual decrease, no clear "elbow"
- Silhouette Score: K=2 achieved highest score (0.1910)

TOO GENERIC

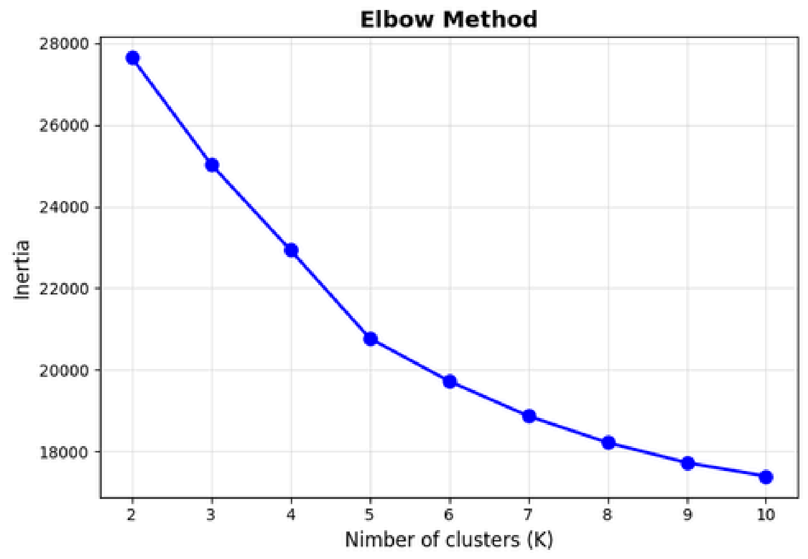
03

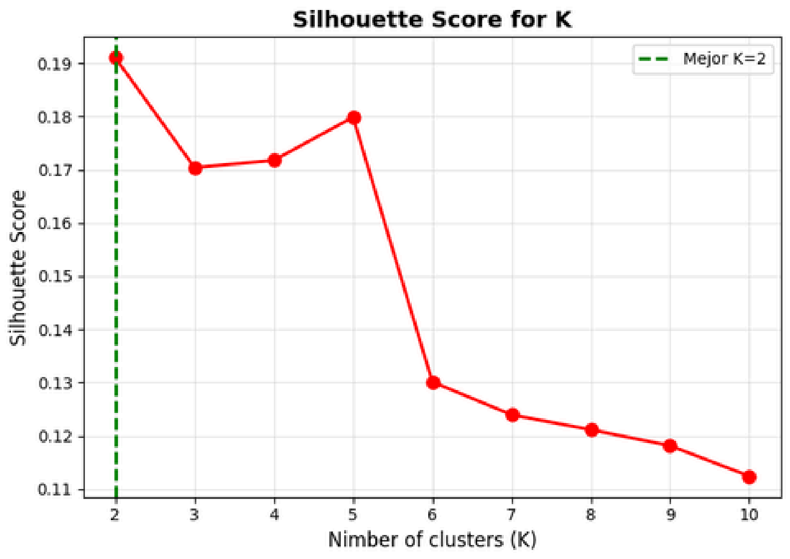
FINAL DECISION

WHY K=3 INSTEAD OF K=2?

Better business segmentation
Balanced distribution

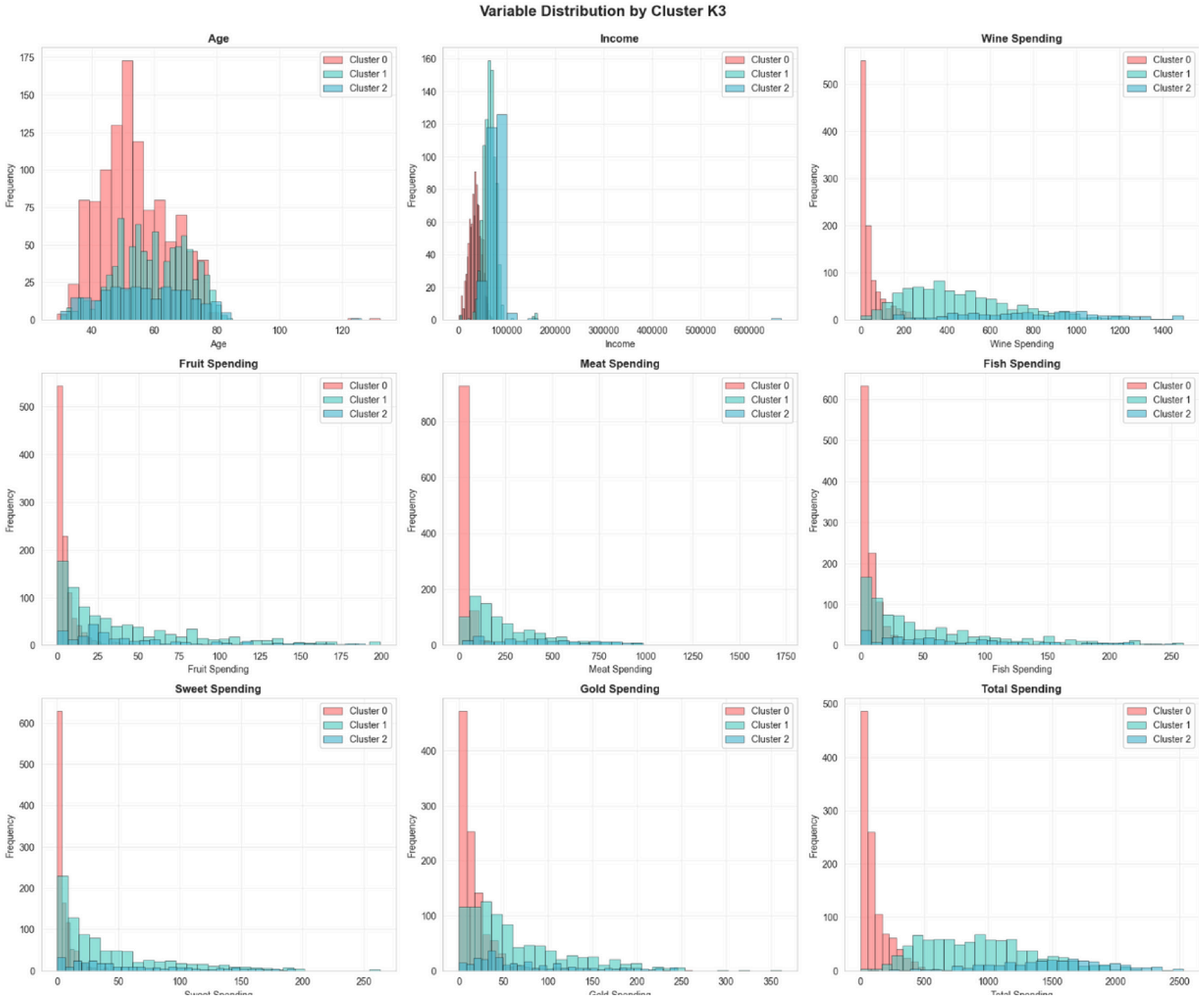
```
K-means with K=3

Distribution of clusters:
Cluster_K3
0    1077
1     889
2     274
Name: count, dtype: int64
```

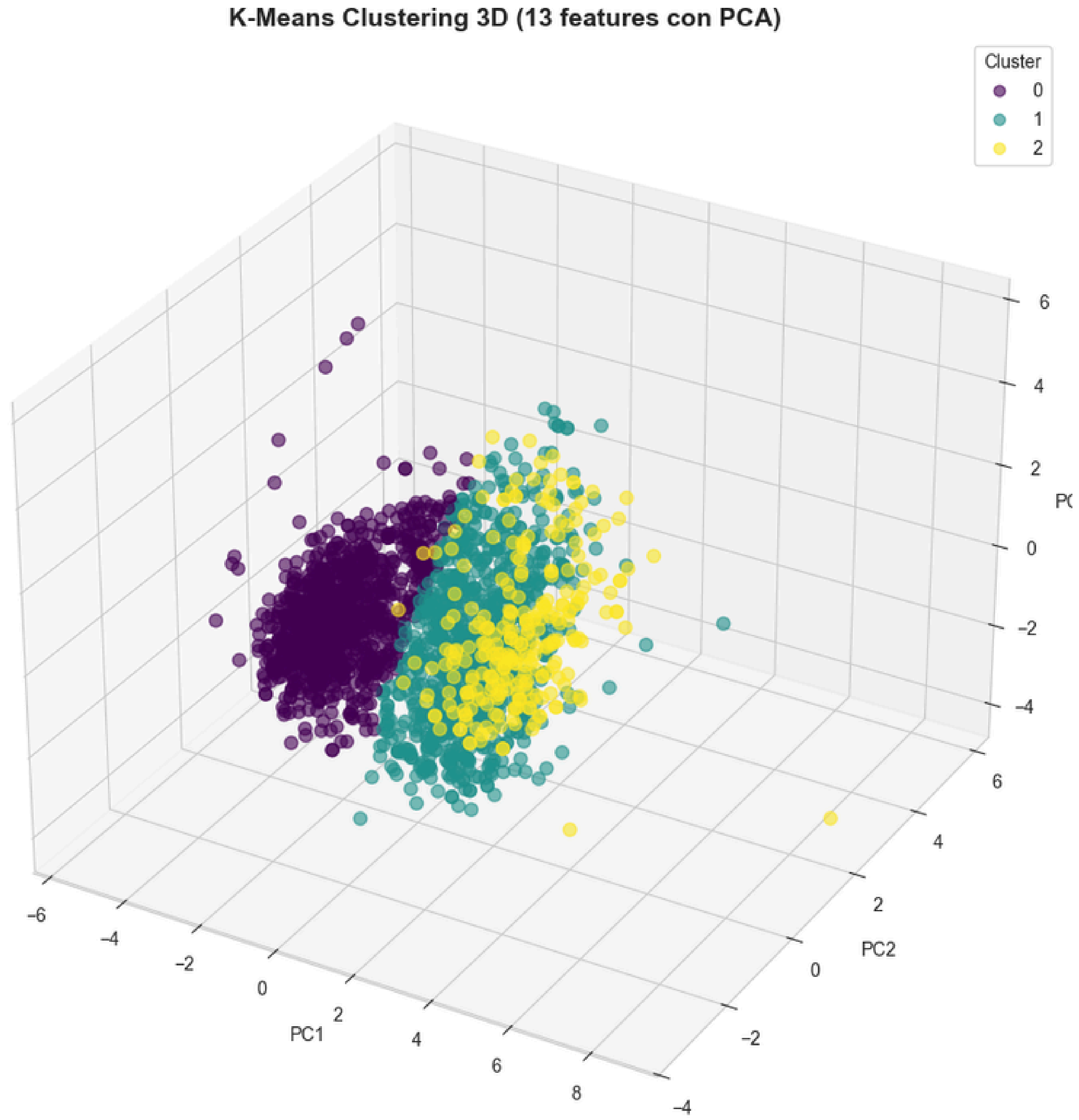


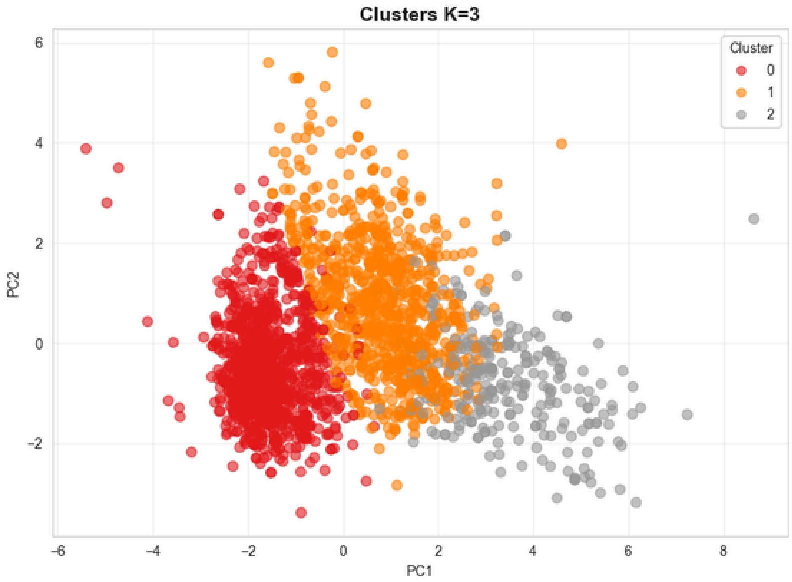


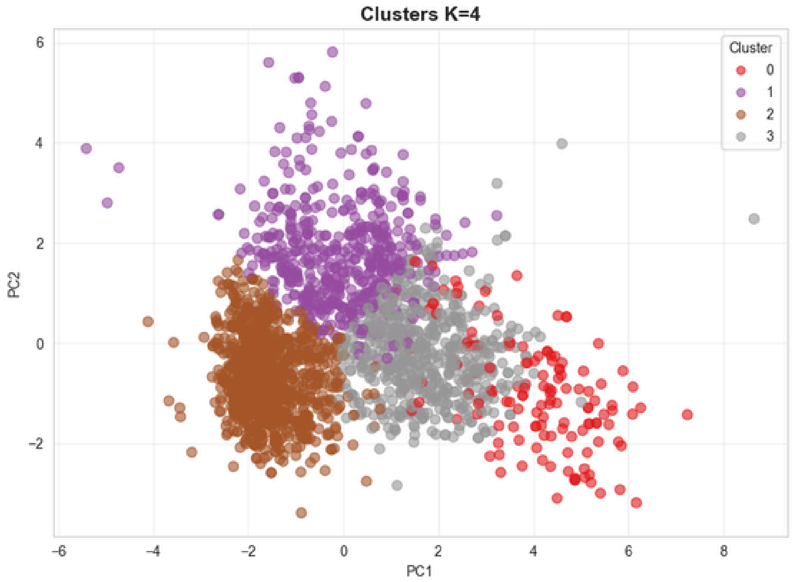
KMEANS_3 = KMEANS(N_CLUSTERS=3, RANDOM_STATE=42, N_INIT=10)

CLUSTERING WITH K-MEANS









CLUSTERING WITH K-MEANS

With the research we got that the best quantity of clusters is K=3, so we get this main differences:

C0: Families aged 53 with low income ($35K) and approximately 1 kid. Minimal spending ($112/year). shop infrequently (8 times/year).

- Basic essentials: Wines ($49), Meat ($27)
- Minimal spend on Fish ($8), Fruits ($5), Sweets ($5), Gold ($18)

C1: Successful older adults (59 years, $65K) with 1 or less. High spending ($929/year), frequent purchases (21 times/year), premium products.

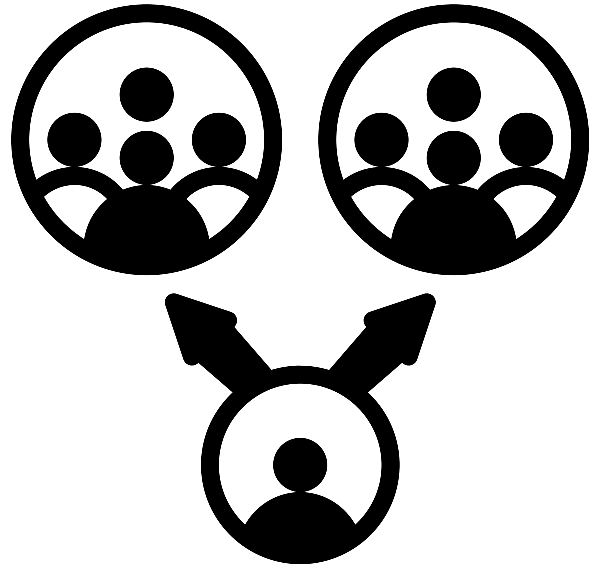
- Premium across board: Wines ($464), Meat ($253) - their favorites
- Significant spend: Fish ($59), Fruits ($43), Sweets ($43), Gold ($66)

C2: Affluent (57 years, $80K) with no kids - maximum purchasing power. Ultra-premium spending ($1,497/year), DON'T use discounts, catalog leaders.

Highest spenders in EVERYTHING:

- Wines ($785)
- Meat ($437)

Other products: Fish ($83), Sweets ($61), Fruits ($54), Gold ($77)



FORECASTING

```python
from statsmodels.tsa.holtwinters import ExponentialSmoothing

forecast_periods = 4
results = {}

print("=== FORECAST WITH DOUBLE EXPONENTIAL SMOOTHING ===")

clusters = {
    0: df_cluster0,
    1: df_cluster1,
    2: df_cluster2
}

for cluster_num, df_cluster in clusters.items():
    print(f"\n--- CLUSTER {cluster_num} ---")

    cluster_results = {}

    for mnt_col in mnt_columns:
        print(f"\nProduct: {mnt_col}")

        X = df_cluster['Observation'].values
        y = df_cluster[mnt_col].values
```

```python
columns_forecast = ['Cluster_K3', 'Dt_Customer', 'MntWines', 'MntFruits', 'MntMeatProducts',
                    'MntFishProducts', 'MntSweetProducts', 'MntGoldProds']
```

DOUBLE EXPONENTIAL SMOOTHING

Moving Average fails to capture trends and reacts slowly to recent changes. ***Simple Exponential Smoothing*** only models the level but ignores trend components. ***Double Exponential Smoothing*** is superior because it captures both the current level and trend direction, providing more accurate multi-period forecasts for time series with fluctuating patterns like wine sales data.

Observations = Months

What we took into account?

```
=== FORECAST WITH DOUBLE EXPONENTIAL SMOOTHING ===

--- CLUSTER 0 ---

Product: MntWines
  RMSE: 889.95
  MAPE: 30.29%
  Number of observations: 23
  Forecast for next 4 months: ['1906.39', '1873.12', '1839.86', '1806.59']

Product: MntFruits
  RMSE: 69.80
  MAPE: 27.25%
  Number of observations: 23
  Forecast for next 4 months: ['150.23', '142.05', '133.87', '125.69']

Product: MntMeatProducts
  RMSE: 572.75
  MAPE: 33.59%
  Number of observations: 23
  Forecast for next 4 months: ['519.16', '450.05', '380.95', '311.84']
```

FORECASTING

```
=== GENERAL STATISTICS ===
Total of models executed: 18/18
RMSE average: 881.32
MAPE average: 27.35%

BEST 3 MODELS (less RMSE):
  Cluster 0 - MntSweetProducts: 65.07
  Cluster 0 - MntFruits: 69.80
  Cluster 0 - MntFishProducts: 101.20

WORST 3 MODELS (biggest RMSE):
  Cluster 2 - MntWines: 3624.65
  Cluster 1 - MntWines: 3169.53
  Cluster 1 - MntMeatProducts: 2259.48

BEST 3 MODELS BY MAPE:
  Cluster 1 - MntWines: MAPE = 15.20%, RMSE = 3169.53
  Cluster 1 - MntGoldProds: MAPE = 18.82%, RMSE = 510.03
  Cluster 1 - MntMeatProducts: MAPE = 19.04%, RMSE = 2259.48
```

RMSE measures absolute error magnitude, while ***MAPE*** shows relative error percentage. Together, they provide a complete view of forecast accuracy - RMSE for error scale and MAPE for interpretability

Cluster	Product	Future_Observation	Forecast
0	MntWines	24	1906.392575
0	MntWines	25	1873.124375
0	MntWines	26	1839.856175
0	MntWines	27	1806.587976
0	MntFruits	24	150.225175
0	MntFruits	25	142.048281
0	MntFruits	26	133.871387
0	MntFruits	27	125.694493
0	MntMeatProducts	24	519.161502
0	MntMeatProducts	25	450.054275
0	MntMeatProducts	26	380.947048
0	MntMeatProducts	27	311.839820
0	MntFishProducts	24	212.600709
0	MntFishProducts	25	199.929753
0	MntFishProducts	26	187.258797

INVENTORY ANALYSIS

	Product	Future_Observation	Forecast
0	MntFishProducts	24	2693.111611
1	MntFishProducts	25	2613.200732
2	MntFishProducts	26	2533.289852
3	MntFishProducts	27	2453.378973
4	MntFruits	24	1962.773390
5	MntFruits	25	1912.946699
6	MntFruits	26	1863.120008
7	MntFruits	27	1813.293316
8	MntGoldProds	24	2593.258529
9	MntGoldProds	25	2452.269056

July
August
September
October

SUM of all clusters

Supplier information	Wines	Fruits	Meat	Fish	Sweet	Gold
Cost per purchase ($)	1500	500	550	550	300	2000
Unit Price ($)	4	2	15	12	2	120
Unit Holding Cost ($/unit/month)	1.25	1.2	1.2	1.2	1.2	10
Store information	Wines	Fruits	Meat	Fish	Sweet	Gold
Inventory Capacity (unit)	50000	100000	100000	100000	100000	5000

DYNAMICAL LOT-SIZING PROBLEM

- Set
  - T set of periods (months)
- Parameters
  - d_t demand in period t
  - p_t unit production cost in period t *(unit price)*
  - s_t setup (order) cost in period t
  - h_t unit holding cost in period t
  - I_{init} initial inventory

  + Ct *inventory capacity*
- Decision variables
  - q_t quantity to produced in period t
  - I_t inventory level at the end of period t
  - y_t binary variable indicating a setup in period t. It is equal to 1 if a setup (order) is active. 0, otherwise.

$$\min \sum_{t \in T} (s_t y_t + p_t q_t + h_t I_t)$$

Subject to:

$$I_0 = I_{init} + q_0 - d_0$$
$$I_t = I_{t-1} + q_t - d_t \qquad \forall t \in T \setminus \{0\}$$
$$q_t \leq M y_t \qquad \forall t \in T$$
$$q_t, I_t \geq 0 \qquad \forall t \in T$$
$$y_t \in \{0, 1\} \qquad \forall t \in T$$

It <= Ct

WINES											
	Month	July	August	September	October						
Demand	d_t	21680	21395	21110	20826			**Inventory Constraint**			
		21680.33	21395.41	21110.477	20825.55			LHS		RHS	
		Cost					July	0	=	0	
Cost per purchase	s_t	1500					August	0	=	0	
Unit holding	h_t	1.25					September	0	=	0	
Unit price	p_t	4					October	0	=	0	
Initial inventory	I_init	0									
Inventory Capacity		50000						**Quantity Constraint**			
								LHS		RHS	
	Month	July	August	September	October		July	21680	<=	85011	
Quantity to buy	q_t	21680	21395	21110	20826		August	21395	<=	85011	
Inventory level (at the end)	I_t	0	0	0	0		September	21110	<=	85011	
Indication of order	y_t	1	1	1	1	*binary*	October	20826	<=	85011	
										M = Sum of demand	
	Total cost per purchase	6000						**Capacity Constraint**			
	Total unit holding	0						LHS		RHS	
	Total unit price	340044					July	0	<=	50000	
							August	0	<=	50000	
Minimize Obj.F.	Total Cost	346044					September	0	<=	50000	
							October	0	<=	50000	

```
---- EQU setup  setup constraint

                  LOWER          LEVEL          UPPER         MARGINAL

july              -INF       -63331.0000          .              .
august            -INF       -63616.0000          .              .
september         -INF       -63901.0000          .              .
october           -INF       -64185.0000          .              .

---- EQU cap  capacity constraint

                  LOWER          LEVEL          UPPER         MARGINAL

july              -INF            .          50000.0000          .
august            -INF            .          50000.0000          .
september         -INF            .          50000.0000          .
october           -INF            .          50000.0000          .

---- VAR q  quantity to produce in period t

                  LOWER          LEVEL          UPPER         MARGINAL

july                .         21680.0000        +INF             .
august              .         21395.0000        +INF             .
september           .         21110.0000        +INF             .
october             .         20826.0000        +INF             .

---- VAR l  inventory level at end of period t

                  LOWER          LEVEL          UPPER         MARGINAL

july                .              .            +INF            1.2500
august              .              .            +INF            1.2500
september           .              .            +INF            1.2500
october             .              .            +INF            5.2500

---- VAR y  binary variable for setup

                  LOWER          LEVEL          UPPER         MARGINAL

july                .             1.0000         1.0000      1500.0000
august              .             1.0000         1.0000      1500.0000
september           .             1.0000         1.0000      1500.0000
october             .             1.0000         1.0000      1500.0000

                            LOWER          LEVEL          UPPER         MARGINAL

---- VAR total_cost          -INF       346044.0000        +INF             .
```

GAMS

FRUITS

	Month	July	August	September	October
Demand	d_t	1963	1913	1863	1813
	Month	July	August	September	October
Quantity to buy	q_t	1963	1913	1863	1813
Inventory level (at the end)	I_t	0	0	0	0
Indication of order	y_t	1	1	1	1
	Total cost per purchase	2000			
	Total unit holding	0			
	Total unit price	15104			
Minimize Obj.F.	**Total Cost**	17104			

SWEET

	Month	July	August	September	October
Demand	d_t	1913	1853	1793	1734
	Month	July	August	September	October
Quantity to buy	q_t	1913	1853	1793	1734
Inventory level (at the end)	I_t	0	0	0	0
Indication of order	y_t	1	1	1	1
	Total cost per purchase	1200			
	Total unit holding	0			
	Total unit price	14586			
Minimize Obj.F.	**Total Cost**	15786			

MEAT

	Month	July	August	September	October
Demand	d_t	11550	11151	10753	10354
	Month	July	August	September	October
Quantity to buy	q_t	11550	11151	10753	10354
Inventory level (at the end)	I_t	0	0	0	0
Indication of order	y_t	1	1	1	1
	Total cost per purchase	2200			
	Total unit holding	0			
	Total unit price	657120			
Minimize Obj.F.	**Total Cost**	659320			

GOLD

	Month	July	August	September	October
Demand	d_t	2593	2452	2311	2170
	Month	July	August	September	October
Quantity to buy	q_t	2593	2452	2311	2170
Inventory level (at the end)	I_t	0	0	0	0
Indication of order	y_t	1	1	1	1
	Total cost per purchase	8000			
	Total unit holding	0			
	Total unit price	1143120			
Minimize Obj.F.	**Total Cost**	1151120			

FISH

	Month	July	August	September	October
Demand	d_t	2693	2613	2533	2453
	Month	July	August	September	October
Quantity to buy	q_t	1963	1913	1863	1813
Inventory level (at the end)	I_t	0	0	0	0
Indication of order	y_t	1	1	1	1
	Total cost per purchase	2200			
	Total unit holding	0			
	Total unit price	90624			
Minimize Obj.F.	**Total Cost**	92824			

Same pattern throughout the six products

CONCLUSION

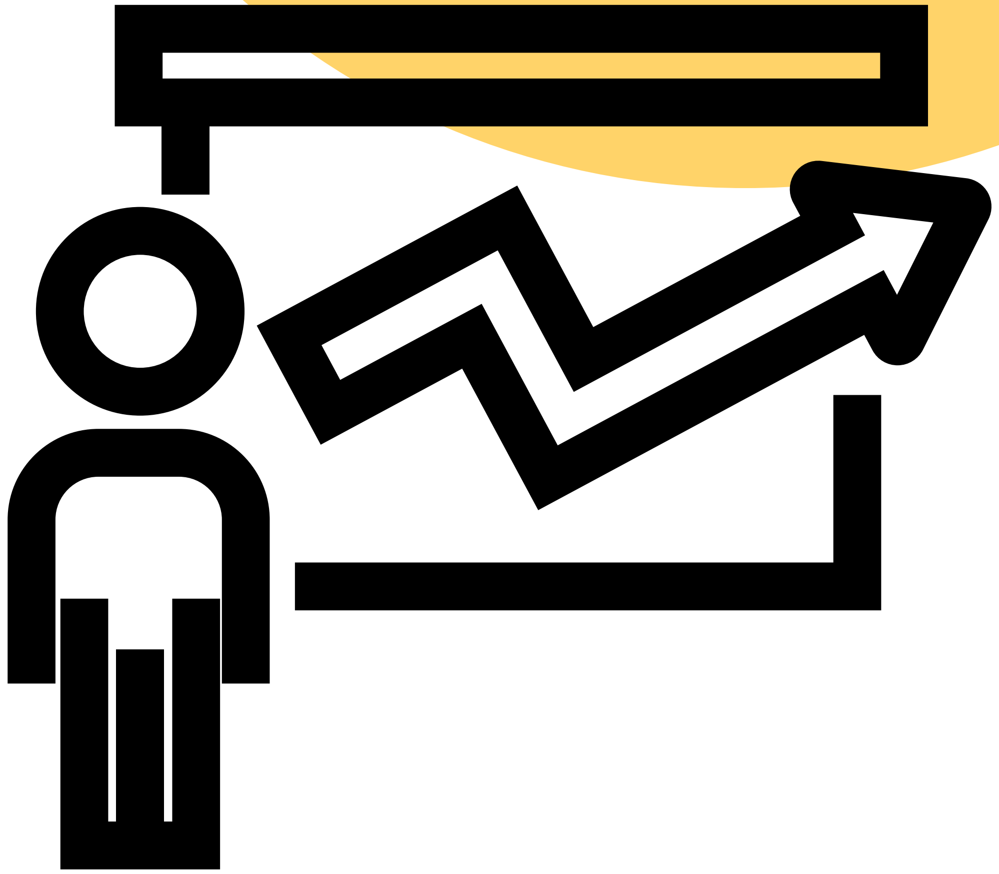
Through the **forecasting, clustering and inventory analysis**, we were able to identify very clearly the customer segments and predict demand trends.

What our analysis revealed was the different customer segments with their respective purchasing behaviors and spending patterns.

Aditionally, implementing Double Exponential Smoothing as a forecasting technique, we were able to provide more accurate sales projections, prevent excess stocking and/or shortages.

Overall, these insights enabled data driven desicion making so the company can be led to a future of better resource allocation, higher customer satisfaction and improved operational efficiency.

By adopting these analytical methods, Nata supermarkets will strenghten their competitive position as well as achive a more sustainable growth.



THANK YOU!