Inventory Optimization

Product and Customer Clustering Based on Demand Patterns

NGUYEN, Duong Tung | Gunwoo Park | Hyeonji Jung
Statistical Data Mining | IMEN472 | Mid-Term Presentation

Outline of Contents

- Introduction
- Data and Preprocessing
- Analysis
- Results
- Roadmap

Introduction

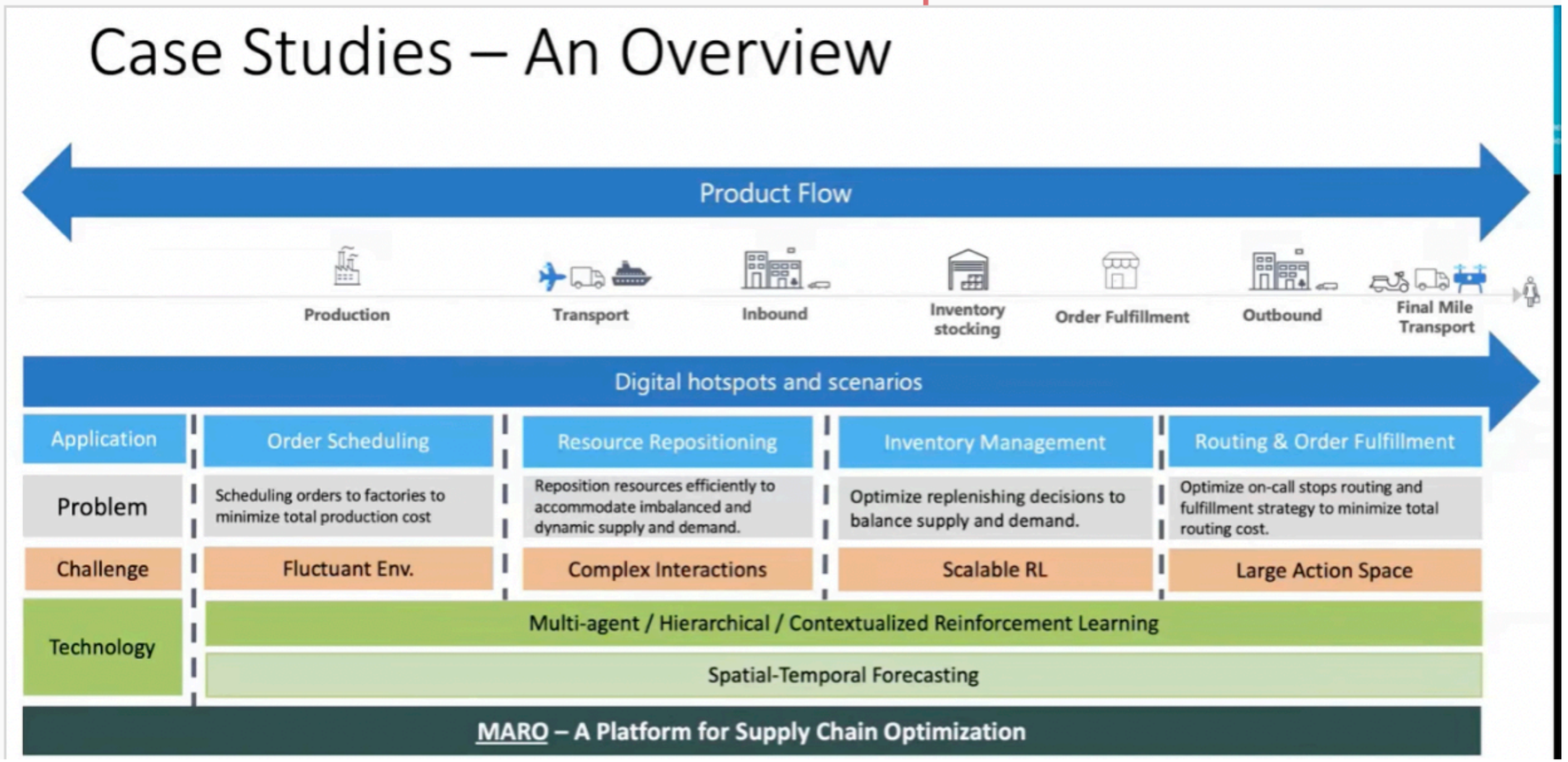
Project Objectives

Domain Background

Introduction - Background

Overview of Supply Chain Processes

[Deep reinforcement learning in SC optimizations: resource optimization - on inventory mgmt](🔗) - Microsoft Research



Introduction - Background

AS-IS: Inventory Management Solutions

[inventory strategies for profit](🔗)

H2 mgmt of inflow, storage, and outflow of inventory

- how well products sell
- keeping the right number of products in stock
- filling orders in an accurate and timely manner
- carefully controlling costs

Overstock / Understock

- product group based adjustments → ABC analysis
  - Optimize inventory turnover
  - reduce obsolete inventory
  - factor into product pricing and supplier negotiations
  - which products to liquify / discontinue
  - identify overvalued / undervalued inventory

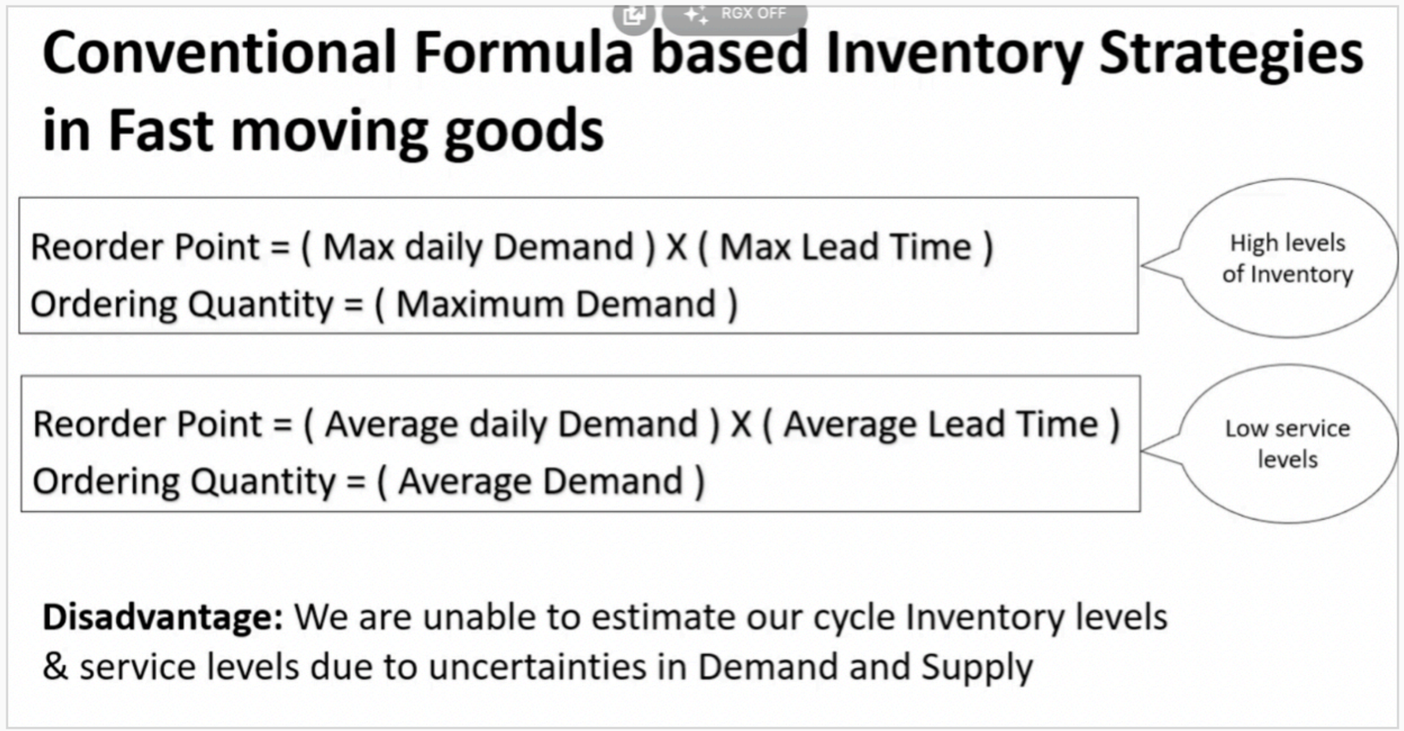
- Ideal stock levels → Economic order quantity(EOQ)
  - ideal amount of inventory to obtain from suppliers in order to meet peak customer demand without running out of stock or tying up too much capital

Introduction - Background

AS-IS: Conventional Inventory Management Solution

[optimize inventory with stochastic simulation , genetic alg](🔗)

As iS:



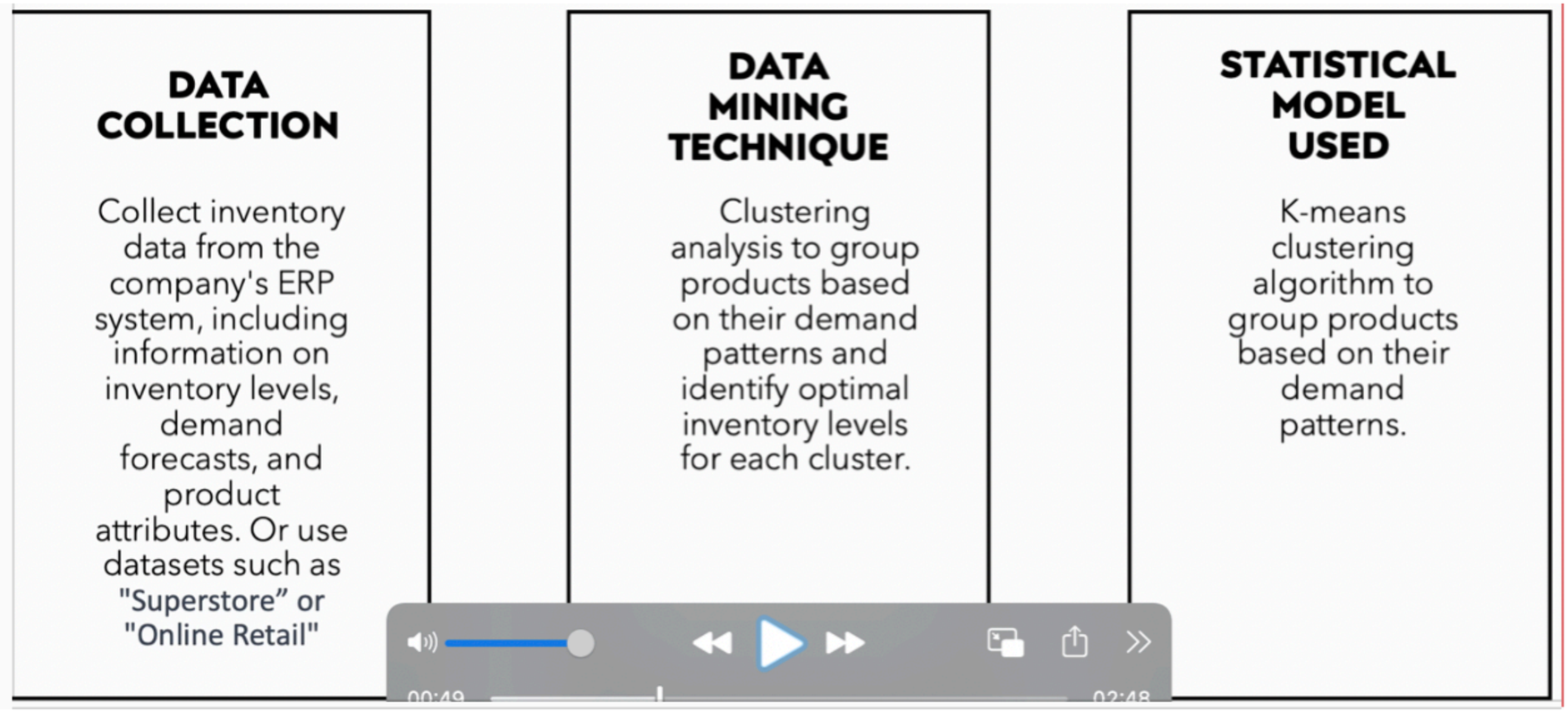
Introduction - Background

AS-IS: Metrics to Avoid

- undesirable stock levels
- late deliveries
- loss of revenue

Introduction - Objective

Overview



Introduction - Objective

Summary of Objectives

The objective of this project is to analyze inventory data from a manufacturing company and cluster products based on their demand patterns to identify optimal inventory levels for each cluster.

- based on attributes(sales, consume rate..) -> categorize -> biz implications
- define product groups with similar demand patterns
- define optimal inventory levels for each cluster based on demand patterns

Introduction - Objective

Implications

EXPECTED RESULT AND BUSINESS/SCIENTIFIC IMPLICATIONS

Expected result	Business/Scientific Implications
1. Groups of products with similar demand patterns that can be managed together to reduce inventory costs and improve supply chain efficiency.	1. Improved inventory management and reduced inventory carrying costs.
2. Optimal inventory levels for each product cluster based on demand patterns and other relevant variables.	2. More efficient use of resources through optimized inventory levels.
	3. Increased revenue through better product availability and faster order fulfillment.

Data

Data Description

Preprocessing

Prep for Feature Selection

Data - Description

Tablueau Superstore Dataset

data for the Sales of multiple products sold by a company

- with subsequent information related to geography
- Product categories, and subcategories, sales, and profits, segmentation amongst the consumers, etc.

used for constructing:

1. monthly profit trend in certain duration
2. expand profit trend to US-states-wide

[Understanding Tableau Superstore Dataset: 4 Important Points](🔗)

```
Tableau Superstore Dataset is a sample Dataset provided by Tableau for new users to learn and experiment with different offerings and functionalities available in Tableau. The vast dataset provides a base to interact with Dashboard, drill down into the specifics of Data handling and provide samples for performing Data transformations.
```

Data - Preprocessing

Setting Objectives Based on Profiling Results

1. Missing Values
2. Variable Type Corrections
3. Encoding
4. Distributions of Variables

Data - Preprocessing

High Cardinality | Sparsity | Duplicate Rows

1. IDs - 4

`Customer ID` has a high cardinality: 793 distinct values	High cardinality
`City` has a high cardinality: 531 distinct values	High cardinality
`Product ID` has a high cardinality: 1862 distinct values	High cardinality
`Product Name` has a high cardinality: 1850 distinct values	High cardinality

➜ assign numeric ids for each original id. reduce every id-related field into one id column.

```
# ASSIGN: ids to string id columns

id_columns = [
    "Customer ID",
    "Product ID",
]
for i in id_columns:
    df[i+'_id'] = df.groupby(i).ngroup()
```

✓ 0.0s

4. Discount - 1

`Discount` has 4798 (48.0%) zeros — Zeros

➜ makes sense since most SKUs would have no discount event. leave it as it is(sparse)

Alerts

Dataset has 2448 (24.5%) duplicate rows — Duplicates

Furniture	Chairs	Global Leather Highback Executive Chair with Pneumatic Height Adjustment, Bla
Furniture	Accessories	Global Leather Highback Executive Chair with Pneumatic Height Adjustment, Bla
Furniture	Appliances	Global Leather Highback Executive Chair with Pneumatic Height Adjustment, Bla
Furniture	Art	Global Leather Highback Executive Chair with Pneumatic Height Adjustment, Bla
Furniture	Binders	Global Leather Highback Executive Chair with Pneumatic Height Adjustment, Bla
Furniture	Bookcases	Global Leather Highback Executive Chair with Pneumatic Height Adjustment, Bla
Furniture	Chairs	Global Leather Highback Executive Chair with Pneumatic Height Adjustment, Bla
Furniture	Copiers	Global Leather Highback Executive Chair with Pneumatic Height Adjustment, Bla

Data - Preprocessing

High Correlations

`Order Date` is highly overall correlated with `Ship Date`	High correlation
`Ship Date` is highly overall correlated with `Order Date`	High correlation
`Sales` is highly overall correlated with `Profit`	High correlation
`Discount` is highly overall correlated with `Profit`	High correlation
`Profit` is highly overall correlated with `Sales` and 1 other fields	High correlation
`Product ID_id` is highly overall correlated with `Category` and 1 other fields	High correlation
`Category` is highly overall correlated with `Product ID_id` and 1 other fields	High correlation
`Sub-Category` is highly overall correlated with `Product ID_id` and 1 other fields	High correlation

Data - Preprocessing

High Correlations

`Sales` is highly overall correlated with `Profit`	High correlation
`Discount` is highly overall correlated with `Profit`	High correlation
`Profit` is highly overall correlated with `Sales` and 1 other fields	High correlation

2. Locations -3

`State` is highly overall correlated with `Postal Code` and 1 other fields	High correlation
`Region` is highly overall correlated with `Postal Code` and 1 other fields	High correlation
`Postal Code` is highly overall correlated with `State` and 1 other fields	High correlation

➜ reduce region-related columns into Region column and drop else

3. SKU category - 2

`Category` is highly overall correlated with `Sub-Category`	High correlation
`Sub-Category` is highly overall correlated with `Category`	High correlation

➜ try to be more careful in product-wise properties, as we're trying to cluster groups of products

Data - Prep for Feature Selection

Boruta: preliminary feature importance check

[Automated feature selection with boruta | Kaggle]( )

Description

Boruta is an all relevant feature selection wrapper algorithm, capable of working with any classification method that output variable importance measure (VIM); by default, Boruta uses Random Forest. The method performs a top-down search for relevant features by comparing original attributes' importance with importance achievable at random, estimated using their permuted copies, and progressively eliminating irrelevant features to stabilise that test.

Details

Boruta iteratively compares importances of attributes with importances of shadow attributes, created by shuffling original ones. Attributes that have significantly worst importance than shadow ones are

```
> print(bor.results)
Boruta performed 100 iterations in 4.965448 mins.
 10 attributes confirmed important: Category, Discount, Postal.Code,
Product.ID, Product.Name and 5 more;
 4 attributes confirmed unimportant: City, Country, Segment,
Ship.Mode;
 4 tentative attributes left: Customer.ID, Customer.Name, Order.Date,
Ship.Date;
```

Data - Prep for Feature Selection

Boruta: preliminary feature importance check

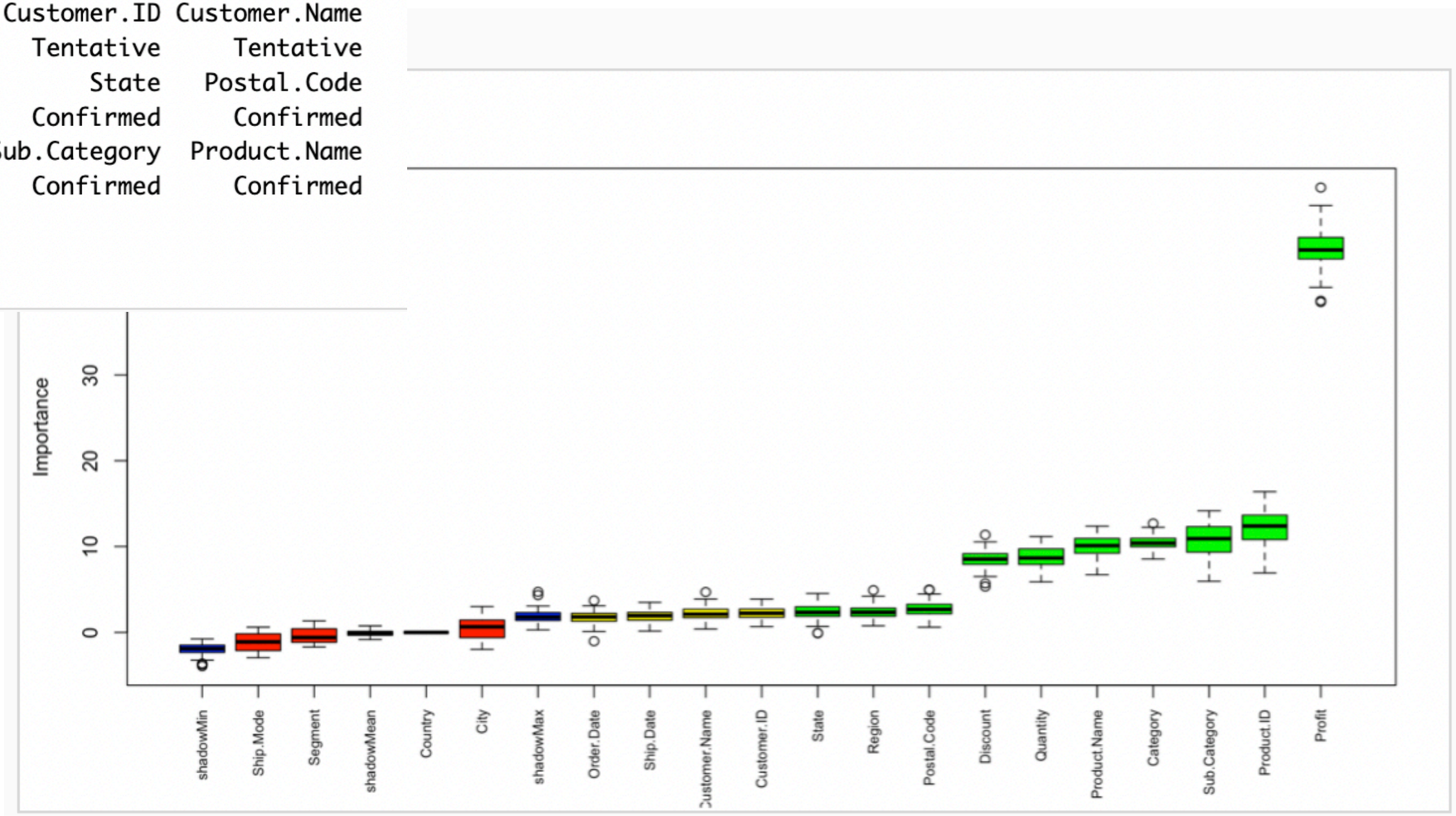
```
> bor.results$finalDecision
   Order.Date      Ship.Date      Ship.Mode    Customer.ID Customer.Name
    Tentative      Tentative       Rejected      Tentative      Tentative
      Segment        Country           City          State    Postal.Code
     Rejected       Rejected       Rejected      Confirmed      Confirmed
       Region     Product.ID       Category Sub.Category   Product.Name
    Confirmed      Confirmed      Confirmed      Confirmed      Confirmed
     Quantity       Discount         Profit
    Confirmed      Confirmed      Confirmed
Levels: Tentative Confirmed Rejected
```

detailed

Description: df [18 × 6]

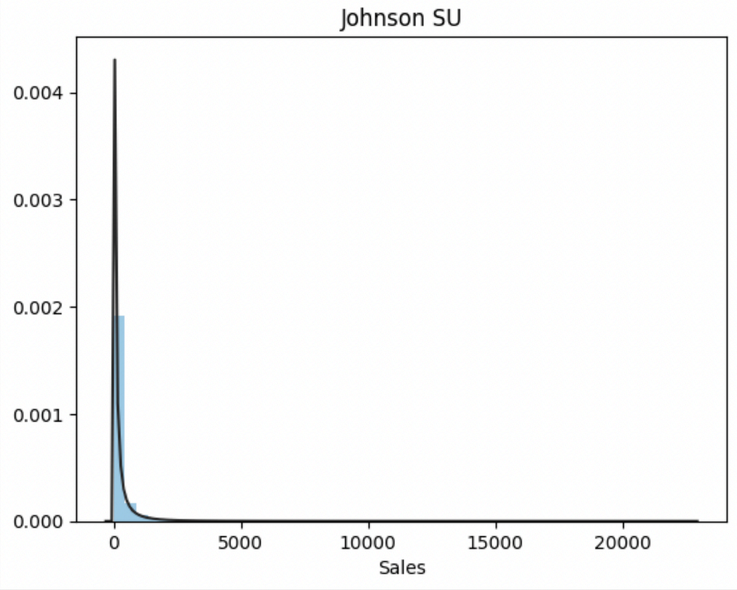
	meanImp <dbl>	medianImp <dbl>	minImp <dbl>	maxImp <dbl>	normHits <dbl>	decision <fctr>
Order.Date	1.7151326	1.7925948	-1.01834424	3.7013969	0.43	Tentative
Ship.Date	1.9306171	1.9424935	0.15422457	3.4918333	0.53	Tentative
Ship.Mode	-1.1604548	-1.1010941	-2.93116284	0.6177136	0.00	Rejected
Customer.ID	2.2472630	2.2292914	0.67763025	3.8772132	0.66	Tentative
Customer.Name	2.2145341	2.1190681	0.39779562	4.6884277	0.62	Tentative
Segment	-0.3459077	-0.5852350	-1.68921652	1.3328082	0.00	Rejected
Country	0.0000000	0.0000000	0.00000000	0.0000000	0.00	Rejected
City	0.3855486	0.6578283	-1.96271562	2.9959233	0.04	Rejected
State	2.3639257	2.3287220	-0.09379559	4.5225128	0.69	Confirmed
Postal.Code	2.7582738	2.6784210	0.61198964	4.9676965	0.75	Confirmed
Region	2.3825293	2.3387804	0.77609549	4.9008364	0.70	Confirmed
Product.ID	12.1910870	12.3780394	6.92009619	16.3674228	1.00	Confirmed
Category	10.4214454	10.3898685	8.54789602	12.6681263	1.00	Confirmed
Sub.Category	10.7240510	10.9097163	5.94314708	14.1466295	1.00	Confirmed
Product.Name	10.0483361	10.0976316	6.72657841	12.3653598	1.00	Confirmed
Quantity	8.6868376	8.6642173	5.87326330	11.1581730	1.00	Confirmed
Discount	8.5104844	8.5391044	5.32658755	11.3597774	1.00	Confirmed
Profit	44.5891772	44.5360875	38.47843839	51.7952812	1.00	Confirmed

18 rows



Data - Distribution of Features

Understanding Feature Characteristics



Sales column has too much Min-Max difference

```
print(df_f['Sales'].min())
print(df_f['Sales'].max())
```

[20] ✓ 0.0s Python

```
0.444
22638.48
```

```
q1 = df["Sales"].quantile(0.01)
q25 = df["Sales"].quantile(0.25)
q75 = df["Sales"].quantile(0.75)
q99 = df["Sales"].quantile(0.99)
print(q1)
print(q25)
print(q75)
print(q99)
```

✓ 0.0s Python

```
2.286
17.28
209.94
2481.694599999993
```

Methods Introduction

Analysis

Rough EDA

Feature Selection

Analysis - Methods
Overview: Clustering Method

H4 clustering

> certain products with high popularity

> start from finding feature importance

> take out specific columns related to specific objectives
> - making groups of columns to certain objective?

1. lack of labeled data
2. ambiguity in cluster labels (vs actual classification labels)
3. limited to specific types of problems(not for predicting continuous)

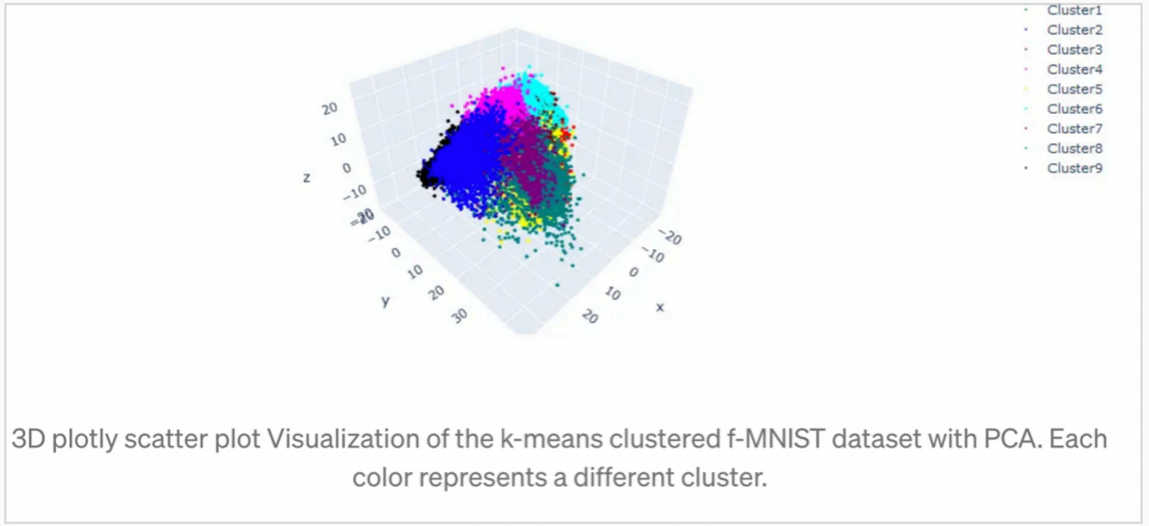
H2 difficulty in interpretations

- what the clusters actually represent?
  - without context/domain knowledge, challenging to determine the meaning of the clusters
- many possible grouping method ➜ which is the most accurate/meaningfu in our case?
- it's a useful exploratory tool to identify pattern - not so clear how to apply the pattern to actual problems or decision making
  1. domain expertise
  2. validation ➜ ensure meaningful clusters, identified patterns arent' random noise, identify bias/limits in the clustering algorithm
     - cross validation, holdout validation(too simple), bootstrapping, external validation, a/b testing
  4. feature engineering ➜ select/tf features so that they are relevant & meaningful for the problem
  5. integration with other techniques
     - set clustering result as the target variable
     - use logistic regression / decision tree to predict the segment for new users

➜ clustering = pattern identification
➜ classification = predict segment for new data

Analysis - Methods
Overview: Analysis Methods

[example of K-means clustered 3d plotly scatter plot](🔗)



3D plotly scatter plot Visualization of the k-means clustered f-MNIST dataset with PCA. Each color represents a different cluster.

[Unsupervised learning algorithms](🔗)

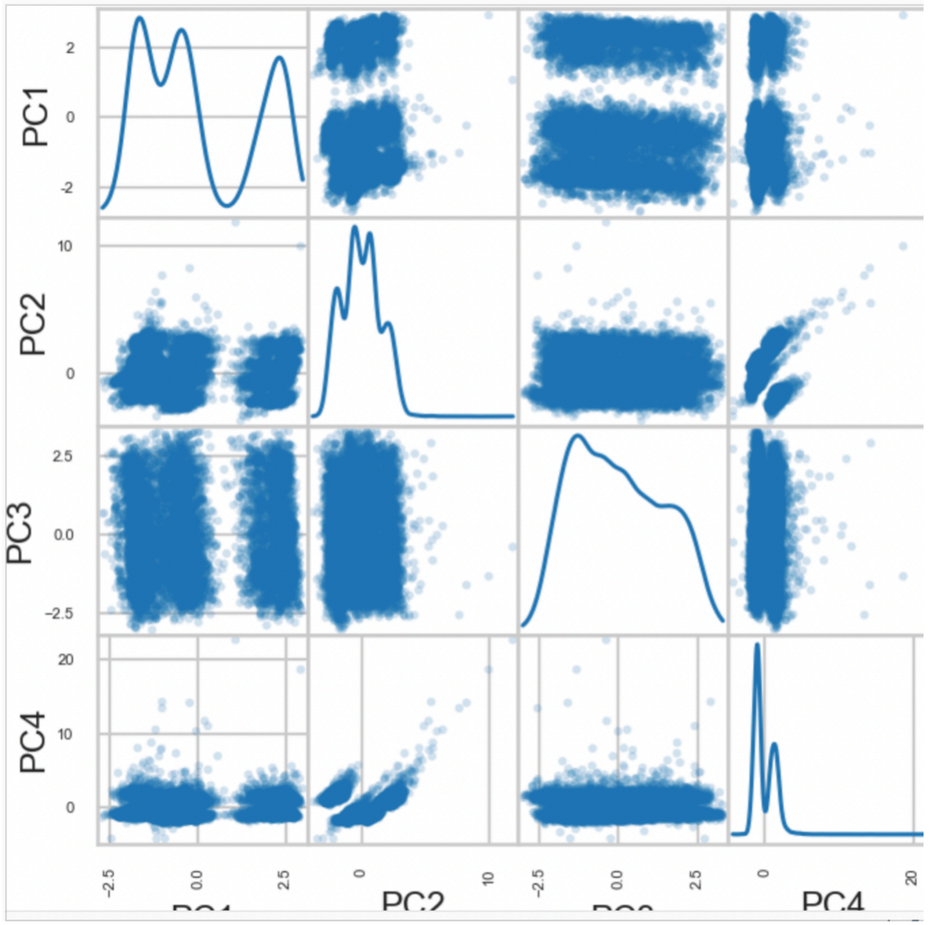
1. understand high dim datasets & visualize → PCA
2. clustering data → KMC
3. improve and validate performance of KMC → elbow, cross validation, performance metrics
4. overcome class overlap → preprocessing with PCA
5. better classification → SOM

SOM(self organizing maps) - dim reduced data representation with topology

[SOM](🔗) or Self-organizing maps 2012 by Teuvo Kohonen

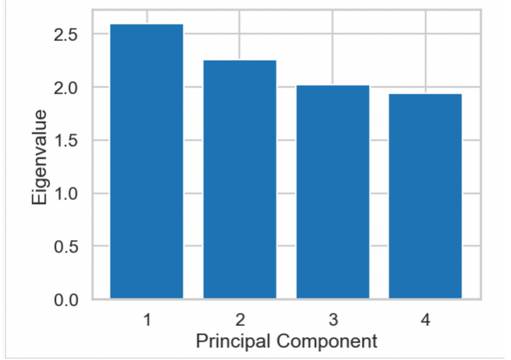
Analysis - Rough EDA

Scatter Plots with PCA Components

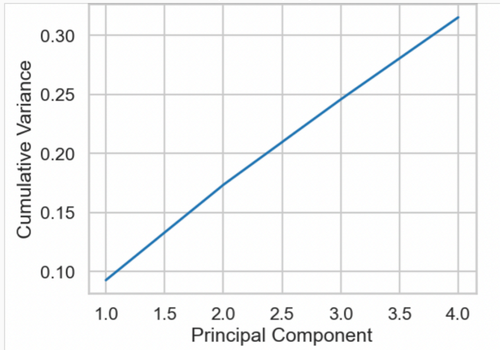


- PC3-PC4, PC3-PC2 : high correlation? - may be a duplicate but interaction with other components could be informative
  - PC3 = subset of PC2, PC4?
- PC4, PC2 : have relatively low variance (less impactful on explaining the dataset?)

checking eigenvalue from PCA - more than 4 significant components seems possible



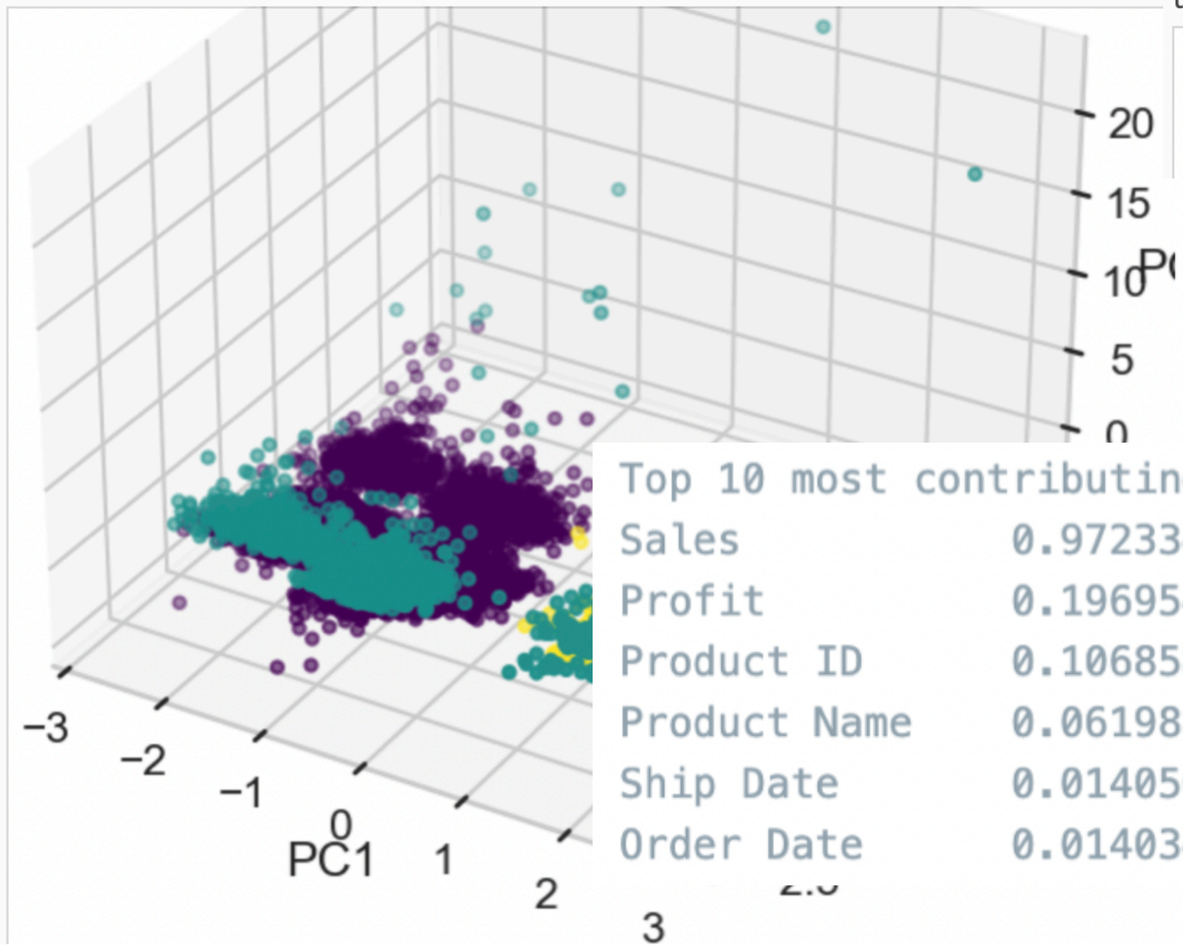
PCA result check - variance is dispersed equally



Analysis - Rough EDA

K-Means Clustering with PCA Components

kmeans with 3 clusters, from pca with 4 components



 # checking for feature-wise contribution to each of the PCs - regardless of the validity of the procedure:

```
Output exceeds the size limit. Open the
Principal Component 1:
Order Date: 0.03153639383224873
Ship Date: 0.03146182541125151
Customer ID: 0.0003962830440504631
```

```
Top 10 most contributing features for PC2:
Sales           0.972334
Profit          0.196954
Product ID      0.106858
Product Name    0.061988
Ship Date       0.014050
Order Date      0.014034
```

```
Top 10 most contributing features for PC3:
Product Name    0.727310
Product ID      0.685379
Sales           0.031715
Profit          0.014256
```

Analysis - With Feature Selection

Product/Customer Segmentation: Demand based on “Sales”

Demand based on sales

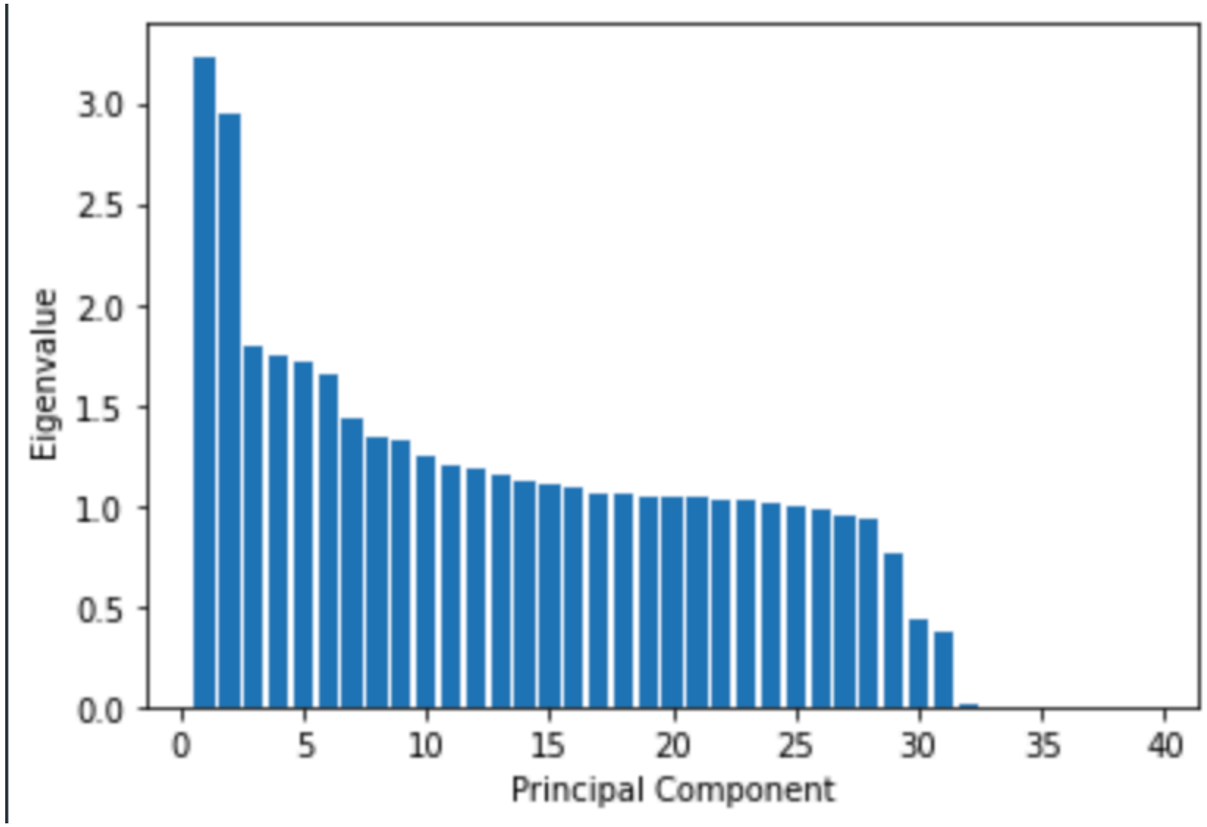
To first make a demand pattern analysis on product, which is to see which product produce the most sales (based on quantity and sales of each order), drop other unrelated columns (based on pre-knowledge).

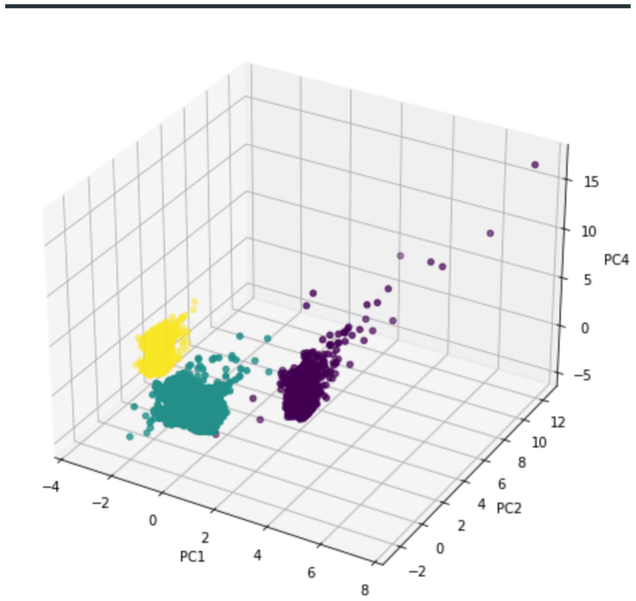
	Product ID	Sales	Quantity
0	FUR-BO-10000112	825.174	9
1	FUR-BO-10000330	1064.624	10
2	FUR-BO-10000362	2154.348	14
3	FUR-BO-10000468	723.842	21
4	FUR-BO-10000711	851.760	12

	Product ID	Sales	Quantity	KMeans_Cluster	DBSCAN_Cluster
0	FUR-BO-10000112	825.174	9	0	0
1	FUR-BO-10000330	1064.624	10	0	0
2	FUR-BO-10000362	2154.348	14	0	0
3	FUR-BO-10000468	723.842	21	0	0
4	FUR-BO-10000711	851.760	12	0	0

Analysis - Model Comparisons

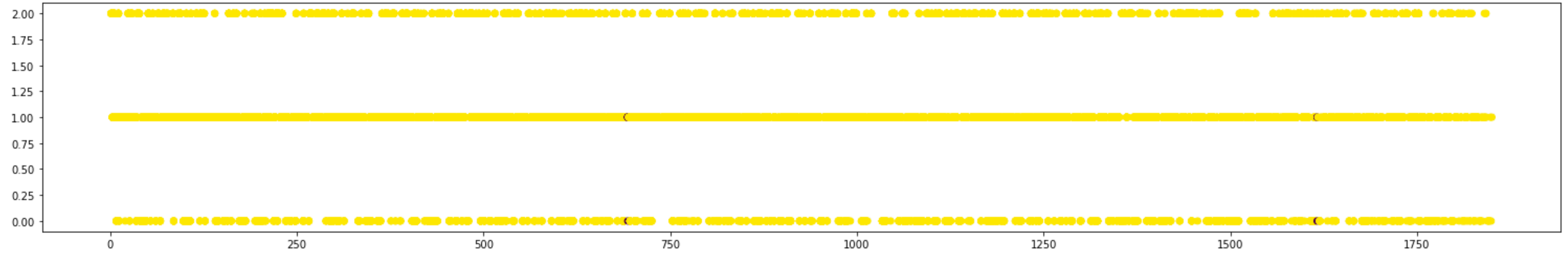
PCA-based clustering

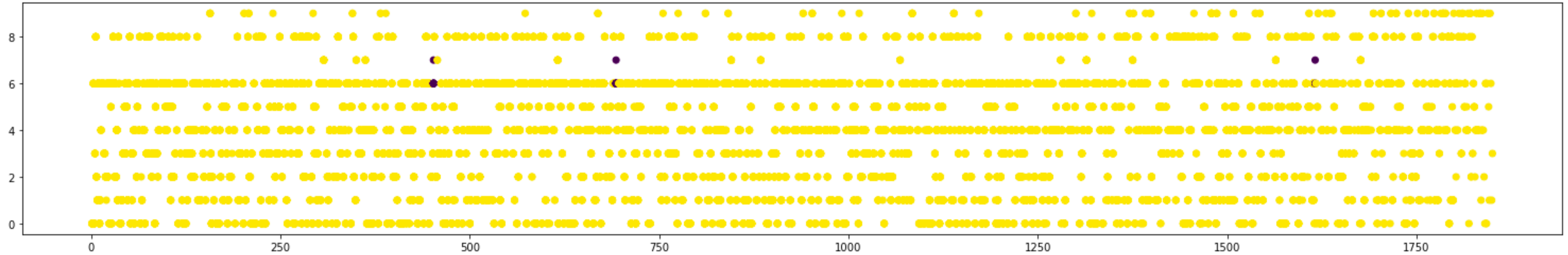




Analysis - Model Comparisons

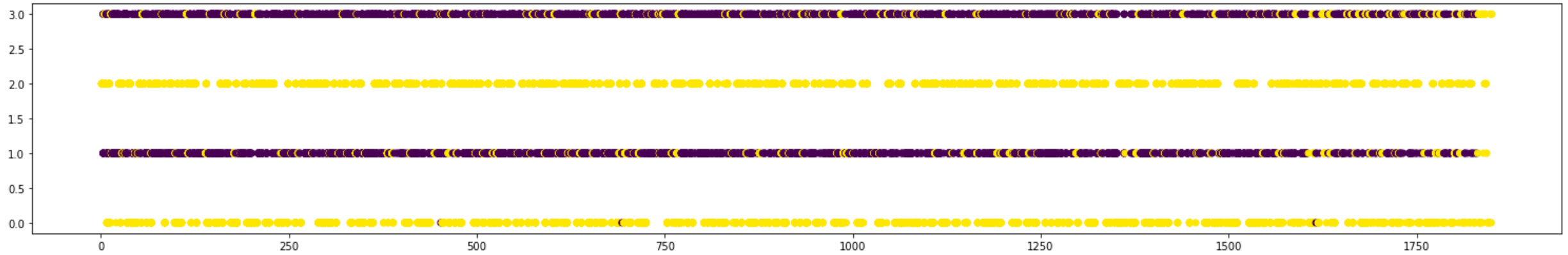
Check for Product Uniqueness Reflected in Cluster Separations

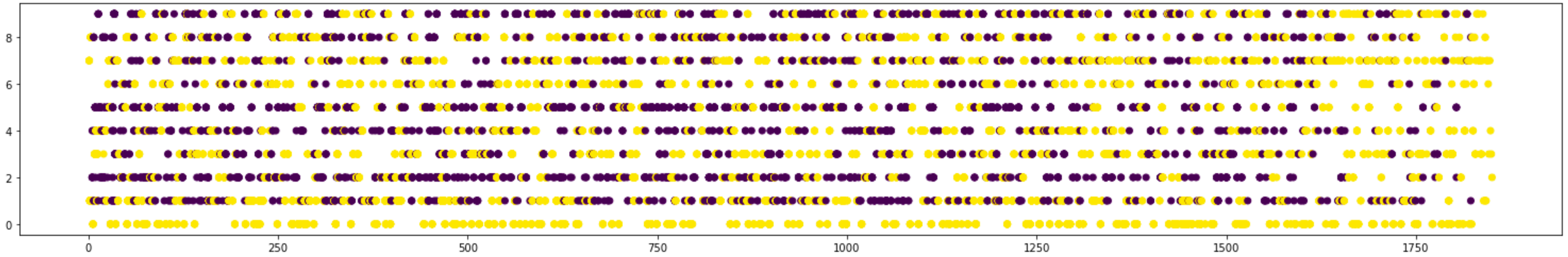




Analysis - Model Comparisons

Check for Product Uniqueness Reflected in Cluster Separations





Results

Rough EDA

Feature-Selected EDA

Model Comparisons

Results - Rough EDA

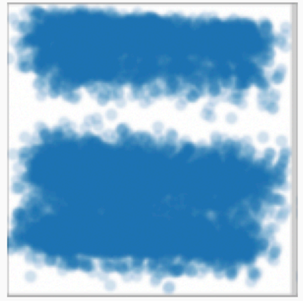
Feature Selection According to PCA-Based Clustering

- **Product Name, Product ID**

```
Top 10 most contributing features for PC3:
Product Name    0.727310
Product ID      0.685379
```

```
Product Name    0.003618
Product ID      0.001449
Order Date      0.001384
Ship Date       0.001384
```

  - Certain products are related to the underlying structure of the dataset (=creates variability) which scores higher in PC1 and PC3
    - although PC1 and PC3 shares the highest variability inducing factor (Product Name/ID), the scatter matrix shows divided group



- **Order/Ship Date**

```
Top 10 most contributing features for PC4:
Order Date      0.706463
Ship Date       0.706407
```

  - PC4 captures variability caused by Order/Ship Date ➜ specific timing patterns could matter according to: Product ID / customer type / seasonality, (and other features)

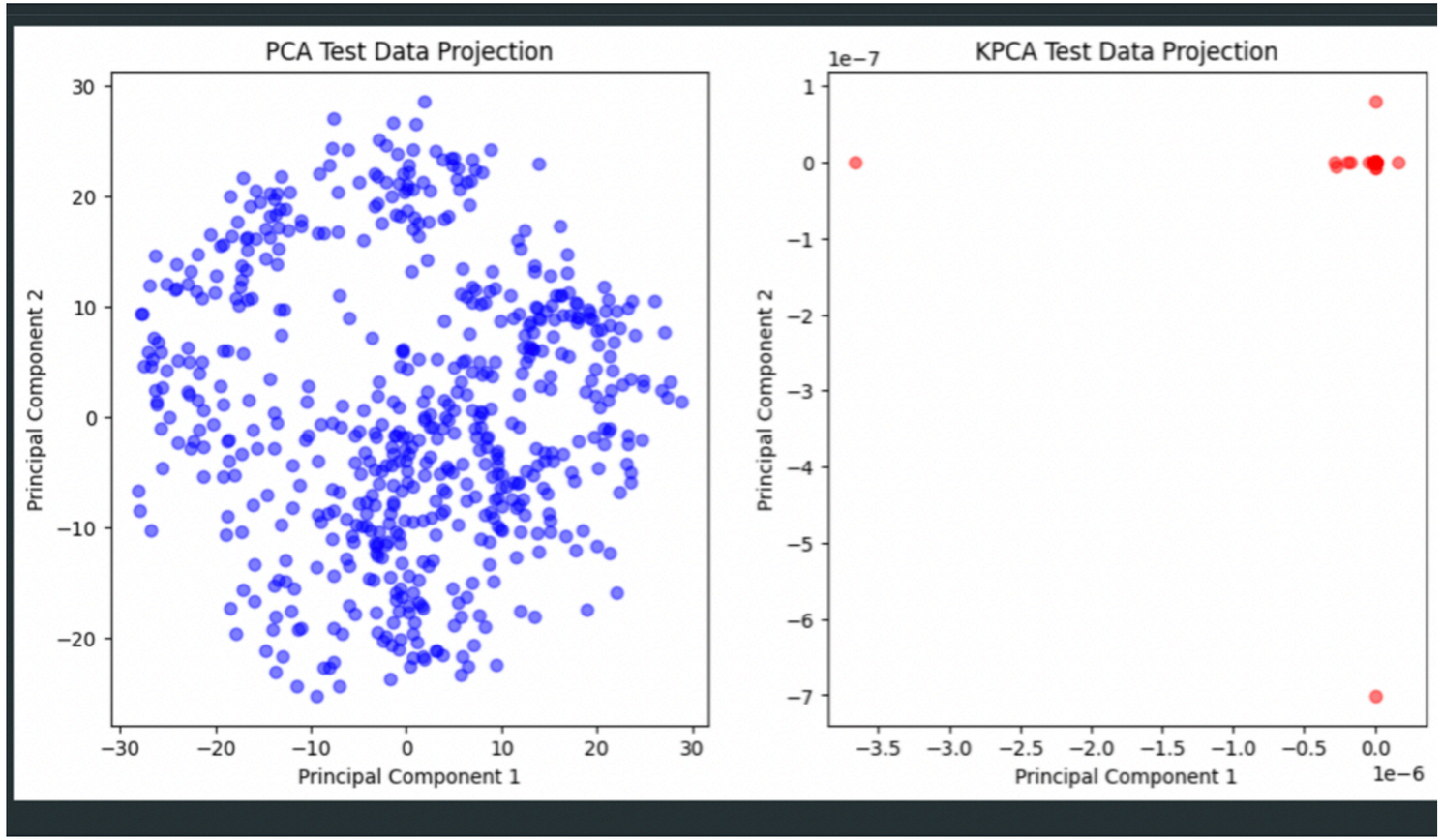
➜ maybe SARIMA onto order/ship date - product sales? - but should need more preprocessing..

- **Sales / Profit**

```
Top 10 most contributing features for PC2:
Sales           0.972334
Profit          0.196954
Product ID      0.106858
Product Name    0.061988
```

Results - Rough EDA

More Preprocessing Required in Scaling Part



Positive Semi Definite problem

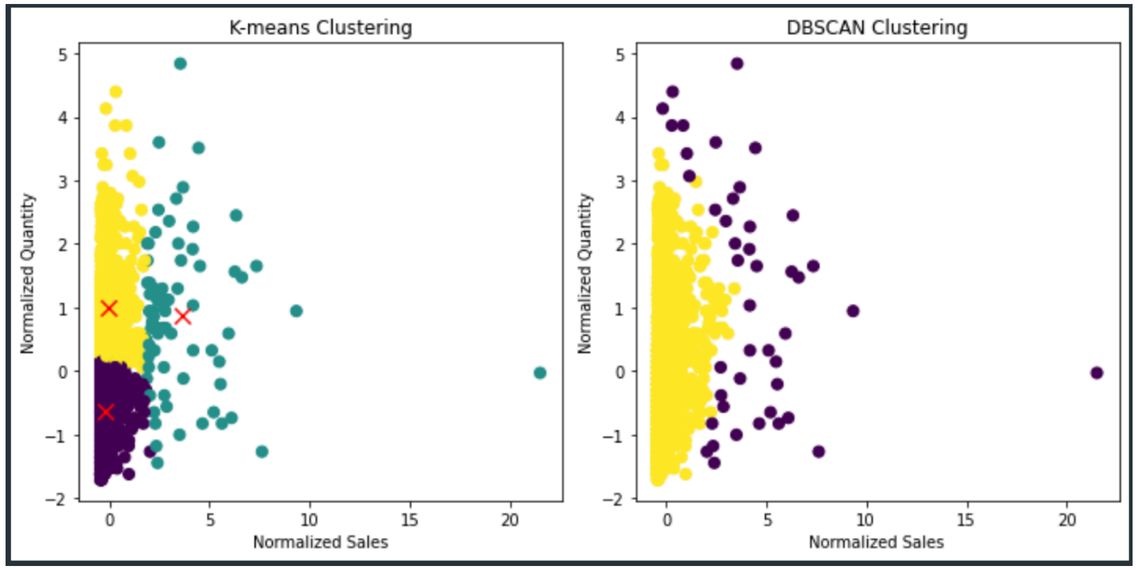
ValueError: There are significant negative eigenvalues (0.105227 of the maximum positive). Either the matrix is not PSD, or there was an issue while computing the eigendecomposition of the matrix.

self._*fit*_transform(K)

test_kpca = kpca.fit(train).transform(test)

Results - FS-EDA

Product: (Sales, Quantity)



Based on the clustering results and the mean Sales and Quantity for each cluster, you can derive some insights for inventory management:

K-means Clusters:

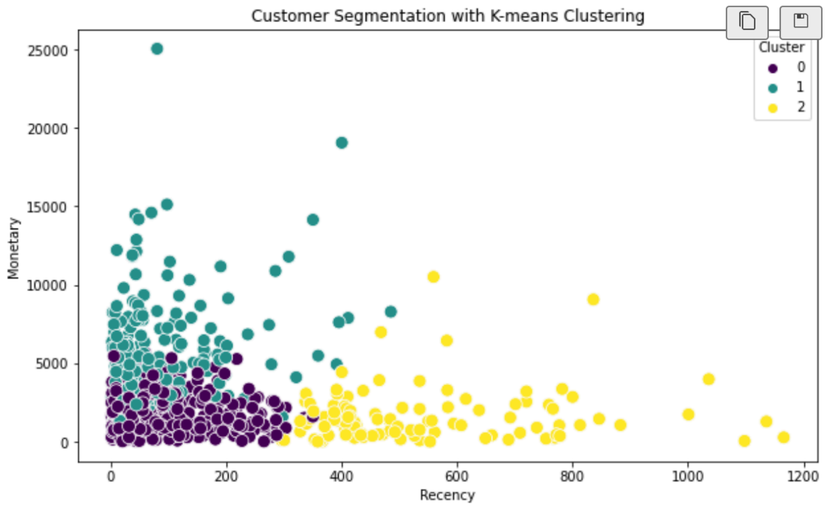
- Cluster 0: These products have relatively low sales (average sales of 631.73) and low quantity (average quantity of 13.24). We can consider this group as low-demand products. For these products, we can keep a smaller inventory and monitor the demand closely to avoid overstocking.
- Cluster 1: These products have high sales (average sales of 11,385.71) and medium quantity (average quantity of 30.26). These are high-demand products that generate significant revenue. It's essential to ensure that you have sufficient inventory for these products to meet customer demand and not miss out on sales opportunities.
- Cluster 2: These products have moderate sales (average sales of 1,130.39) and high quanti (average quantity of 31.59). These products can be considered as medium-demand products. K-means Cluster 2 has a high average quantity (31.59) but relatively lower sales (1,130.39) compared to Cluster 1. This indicates that products in Cluster 2 may have a lowe price per unit or sell in larger quantities per transaction than those in Cluster 1. Implement inventory optimization techniques, such as reorder point and economic order quantity (EOQ), to minimize inventory costs and maintain an optimal stock level for these products. Review the pricing strategy for these products. If they have a lower profit margin due to lower price per unit, you might consider adjusting the pricing to increase profitability, provided it doesn't negatively impact demand. Might be able to negotiate better deals with suppliers for products in Cluster 2, given their higher average quantity. This can help reduc the cost per unit and improve the overall profitability of these products.

DBSCAN:

- Cluster -1 (Noise points): These products have very high sales (average sales of 13,245.40) and high quantity (average quantity of 34.20). This cluster consists of products that do not fit well into the other clusters, which might represent outliers or unique products with specific demand patterns. These outliers could have more analysis and might be able to increase stocking of these products
- Cluster 0: These products have moderate sales (average sales of 943.02) and moderate quantity (average quantity of 20.00). This group can be considered as medium-demand products. For these products, you can maintain a balanced inventory level and closely monitor the demand trends to adjust inventory accordingly.

Results - FS-EDA

Customer Segmentation

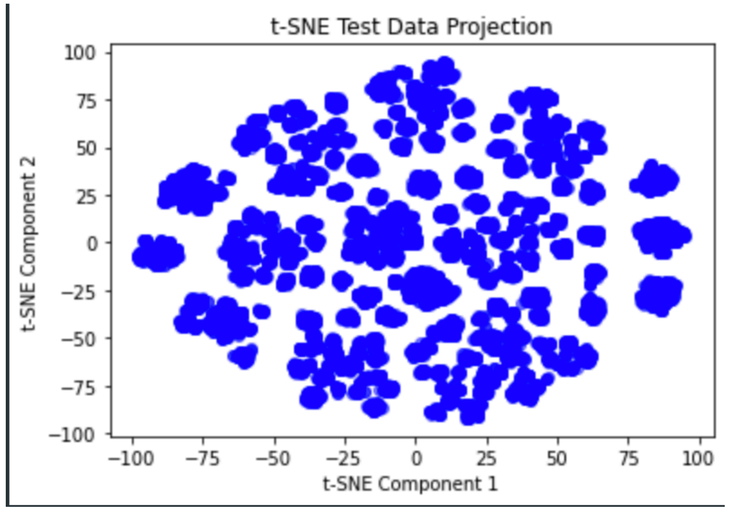


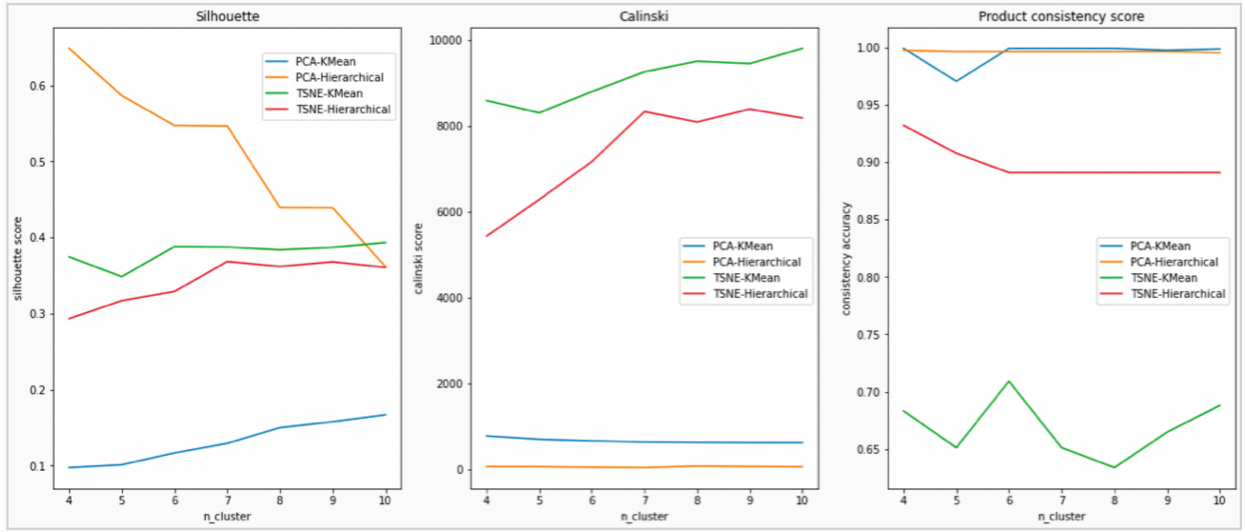
We see that we can get some insights from these customers cluster and have some insights based on these clusters:

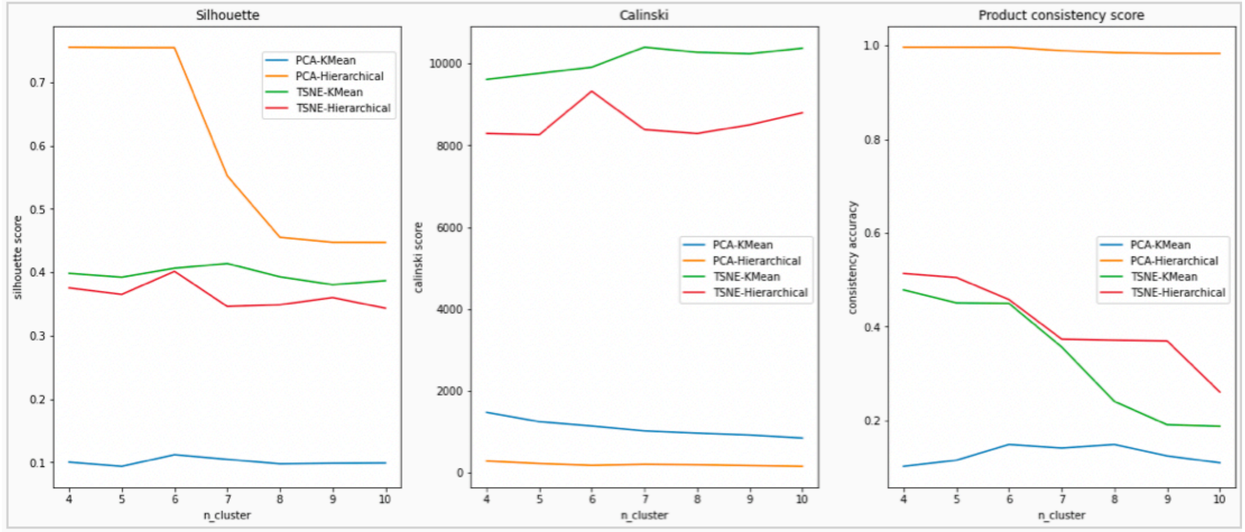
- Cluster 0 is Low recency and low monetary: Insights: These customers have made purchases recently but haven't spent much on their purchases. They might be new or occasional customers who are still evaluating your products or services. Actions: Encourage repeat purchases by offering personalized deals or promotions based on their past purchases or browsing history. Nurture these customers through targeted email campaigns or retargeting ads to build brand awareness and familiarity. Provide excellent customer service and support to convert them into loyal customers. Monitor their behavior and preferences to better understand their needs and interests, and use this information to tailor your offerings.
- Cluster 1 is high monetary and low recency: Insights: These customers have made purchases recently and contributed significant revenue to the business. They are likely to be your most valuable and loyal customers. Actions: Prioritize customer retention and maintain strong relationships with these customers through personalized communication and tailored offers. Offer loyalty program benefits or exclusive rewards to encourage continued engagement and spending. Gather feedback from these customers to understand their preferences and identify opportunities for improvement in products or services.
- Cluster 2 is high recency and high monetary: Insights: These customers have spent a significant amount of money in the past but haven't made any recent purchases. They might be inactive or at risk of churning. Actions: Implement re-engagement strategies, such as sending personalized offers, reminders, or win-back campaigns to bring them back to your business. Identify the reasons for their inactivity, such as dissatisfaction with a product or service, and address these issues. Offer incentives or special deals to encourage them to make a purchase and re-establish their connection with your business. Monitor the success of your re-engagement efforts and adjust your strategies accordingly.

Results - Model Comparisons

By Performance Metrics







Roadmap Evaluation

H_2 for final pt

- more focus on the implications of the statistical results
  - possibly with simulation with past data and our classfication model, validating performance in terms of business aspects as well as statistical significance
- it feels like we're going to need to raise valid reasons for each step in the series of our decisions.
  - what combination of variables / silmilarity measures / clustering techniques led to the classifications
  - altering variables, choosing different similarity measure, concentrating on a particular subset
  - concerns about evaluation of clustering solutions
    - are the clusters real or arbitrary?
    - what other solutions are possible and which is better?
  - interpretation for how each clusters are grouped in the way they are?

refer: Dubes and Jain, 1979

Seasonal Trends

Roadmap Evaluation

```python
# Investigate seasonal trends in sales and quantity
monthly_sales = df.groupby(['Year', 'Month'])['Sales'].sum().reset_index()
sns.lineplot(data=monthly_sales, x='Month', y='Sales', hue='Year', marker='o')
plt.show()
```

Python