STIBO SYSTEMS
MASTER DATA MANAGEMENT

BUSINESS PROPOSAL

CONNECTING ALL OF US WITH THE WORLD

CROSS-SELL AND UPSELL RECOMMENDATION ENGINE

WE NEED TO RECOMMEND MORE PRODUCTS TO MAKE THE CUSTOMER EXPERIENCE BETTER

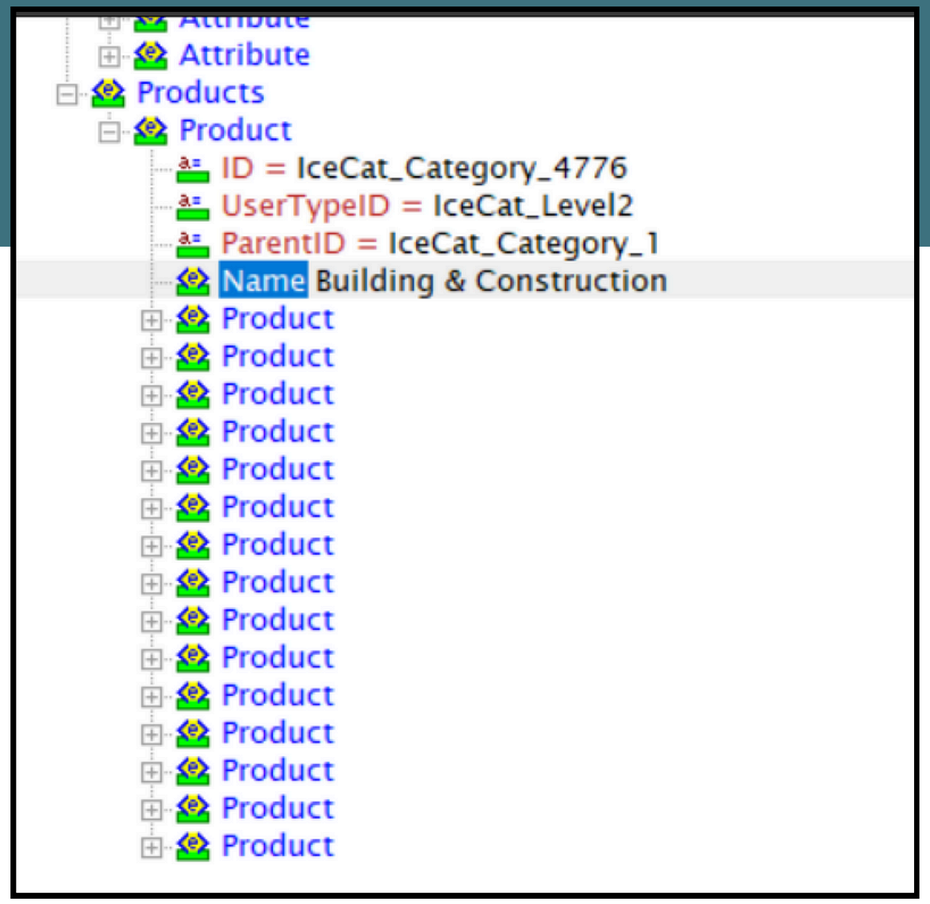
WHAT ARE WE TRYING TO SOLVE

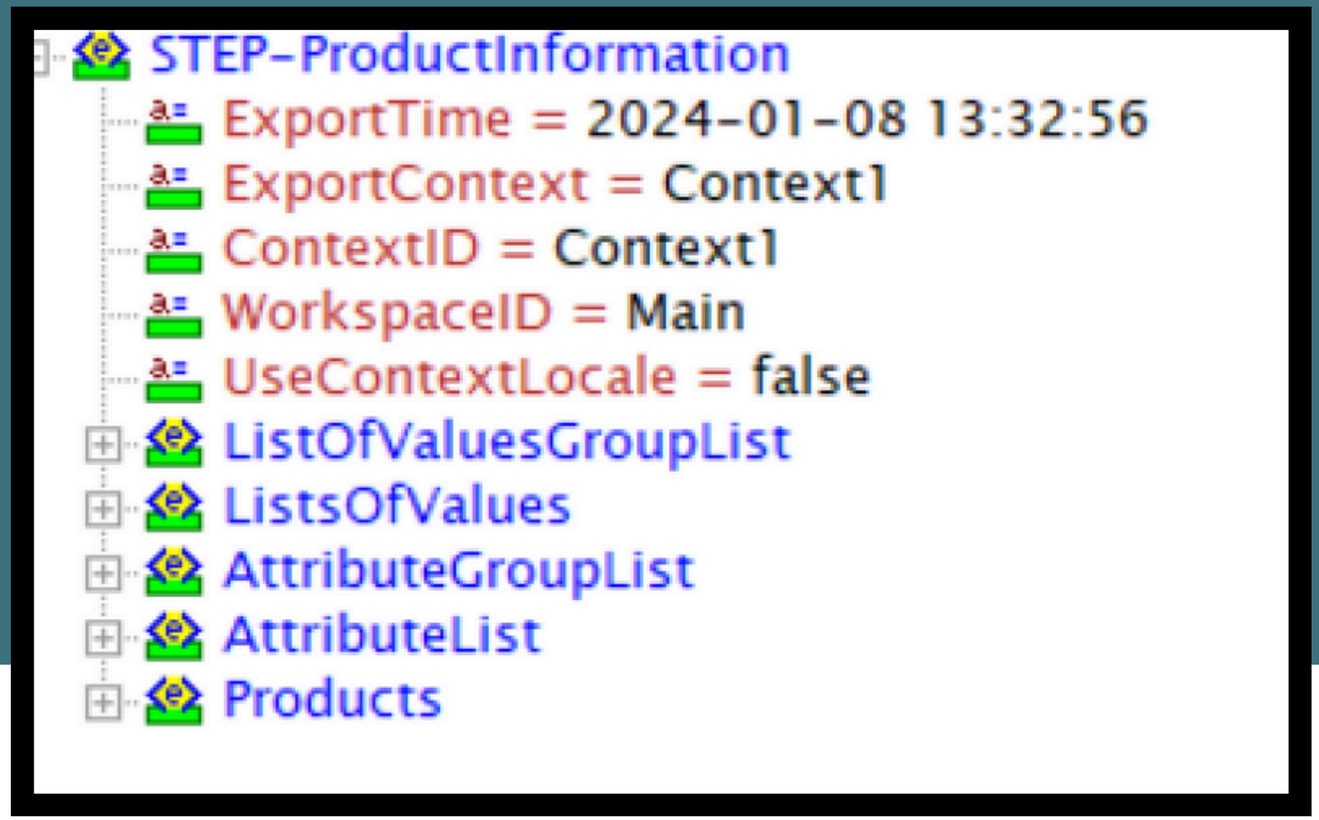
- BY SUGGESTING COMPLEMENTARY OR UPGRADED PRODUCTS DURING THE SHOPPING JOURNEY

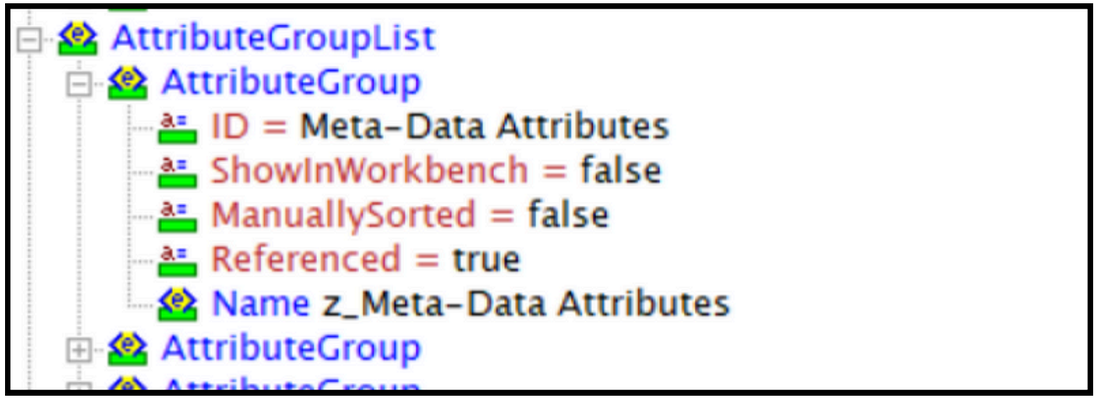
- MAXIMISE REVENUE
- ENHANCE CUSTOMER EXPERIENCE
- INCREASE CUSTOMER LOYALTY

HOW?

XML DATASET


Attribute
Products
Product
ID = IceCat_Category_4776
UserTypeID = IceCat_Level2
ParentID = IceCat_Category_1
Name Building & Construction
Product
Product
Product
Product
Product
Product
Product
Product
Product
Product
Product
Product
Product
Product
Product
Product
Product


STEP-ProductInformation
ExportTime = 2024-01-08 13:32:56
ExportContext = Context1
ContextID = Context1
WorkspaceID = Main
UseContextLocale = false
ListOfValuesGroupList
ListsOfValues
AttributeGroupList
AttributeList
Products


AttributeGroupList
AttributeGroup
ID = Meta-Data Attributes
ShowInWorkbench = false
ManuallySorted = false
Referenced = true
Name z_Meta-Data Attributes
AttributeGroup

PRODUCT DATA

Purchased Together Group	Items Bought Together
Home & Kitchen	"Kitchen Equipment LOVs", "Kitchen Merchandise", "Plumbing/Heating/Ventilation/Air Conditioning"
Food & Beverages	"Vegetables", "Meat/Poultry", "Prepared/Preserved Foods", "Oils/Fats Edible"
Gardening	"Lawn/Garden Supplies", "Shrubs/Trees", "Seeds/Spores", "Seedlings - Ready to Grow"
Health & Wellness	"Medical Devices", "Skin Products", "Pharmaceutical Drugs", "Pet Care"
Pets	"Pet Care/Food", "Pet Care/Food Variety Packs", "Veterinary Healthcare"
Entertainment	"Music", "Musical Instruments/Accessories", "Toys/Games", "Photography/Optics"
Safety & Security	"Safety/Protection - DIY", "Safety/Security/Surveillance", "Lubricants/Protective Compounds"
Office & Stationery	"Stationery/Office Machinery", "Tool Storage/Workshop Aids", "Textual/Printed/Reference Materials"
Electronics	"In-car Electronics", "SmartHomeLOVs", "Tablet LOVs", "VehicleLoVs"
Fashion & Personal Care	"Personal Accessories", "Personal Hygiene Products", "Personal Safety/Security"
Outdoor Sports	"Sports Equipment", "Fishing Gear", "Camping Accessories"
Food Preparation	"Ingredient Smart Tracker", "Ingredient Smart Tracker LOV", "Seasonings/Preservatives/Extracts"
Sustainability & Eco	"Sustainability LOVs", "Life Cycle Assessments", "Insect/Pest/Allergen Control"
Construction & DIY	"Tools/Equipment - Hand", "Tools/Equipment - Power", "ScaffoldingLOVs"
Toys & Kids	"Toys/Games", "ToyLoVs", "Personal Intimacy"

Take your PIM to the next level with the right master data management solution for your industry

CROSS-SELL AND UPSELL RECOMMENDATION ENGINE

1

MAXIMISE REVENUE

We do this by UPSELLING our products

2

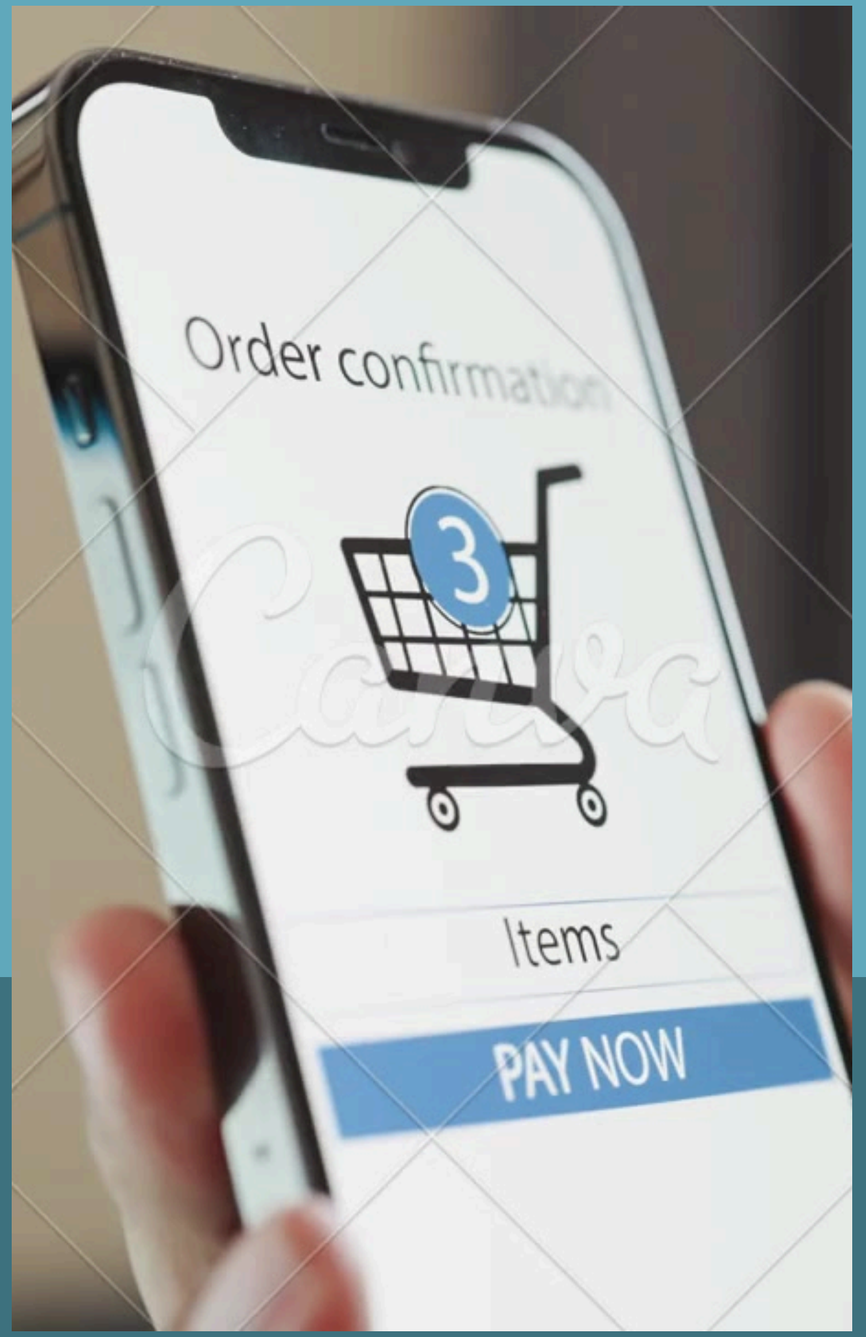
ENHANCE CUSTOMER EXPERIENCE

By CROSS-SELLING, like bread with butter, diapers with baby formula etc

3

INCREASE CUSTOMER LOYALTY

Personalising these recommendations



SOLUTION -1

BUNDLED PRODUCTS

- *Encourage customers to purchase multiple products by offering product bundles at a discounted price.*
- *Create bundles that include the main products along with complementary items.*
- *Highlight the savings and the value customers will get by purchasing the bundle.*

OFFICE ITEMS

FITNESS COMBOS

SOLUTION-2

SEASONAL AND TREND ANALYSIS

- *Analyze seasonal trends and customer behavior during specific periods. Adjust recommendations to align with seasonal demands or events.*
- *Incorporate trending products or popular items into your recommendations to capitalize on current market interests.*

SEASONAL VEGETABLES

BIG SALE

GET UP TO **50% OFF**

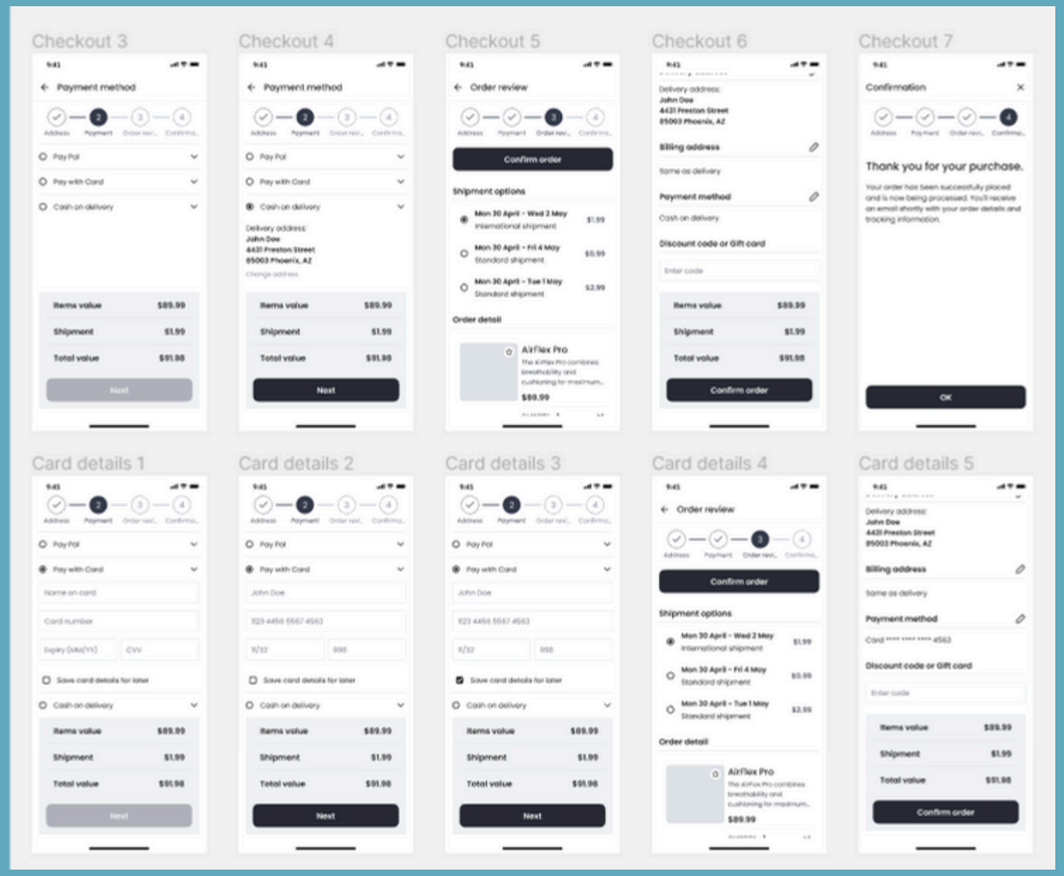
SUMMER COMBOS

SOLUTION-3

MULTICHANNEL RECOMMENDATIONS:

- *Extend recommendations beyond the online store to other channels, such as email, mobile apps, or social media.*
- *Ensure a consistent and coherent recommendation experience across various touchpoints.*

PHONE APP WIREFRAME



SOLUTION-4

CHATBOTS FOR PERSONALISED

- *Chatbots for user interaction and real time data collection*
- *It can tailor the recommendations system for better selling of products*
- *LLM models*

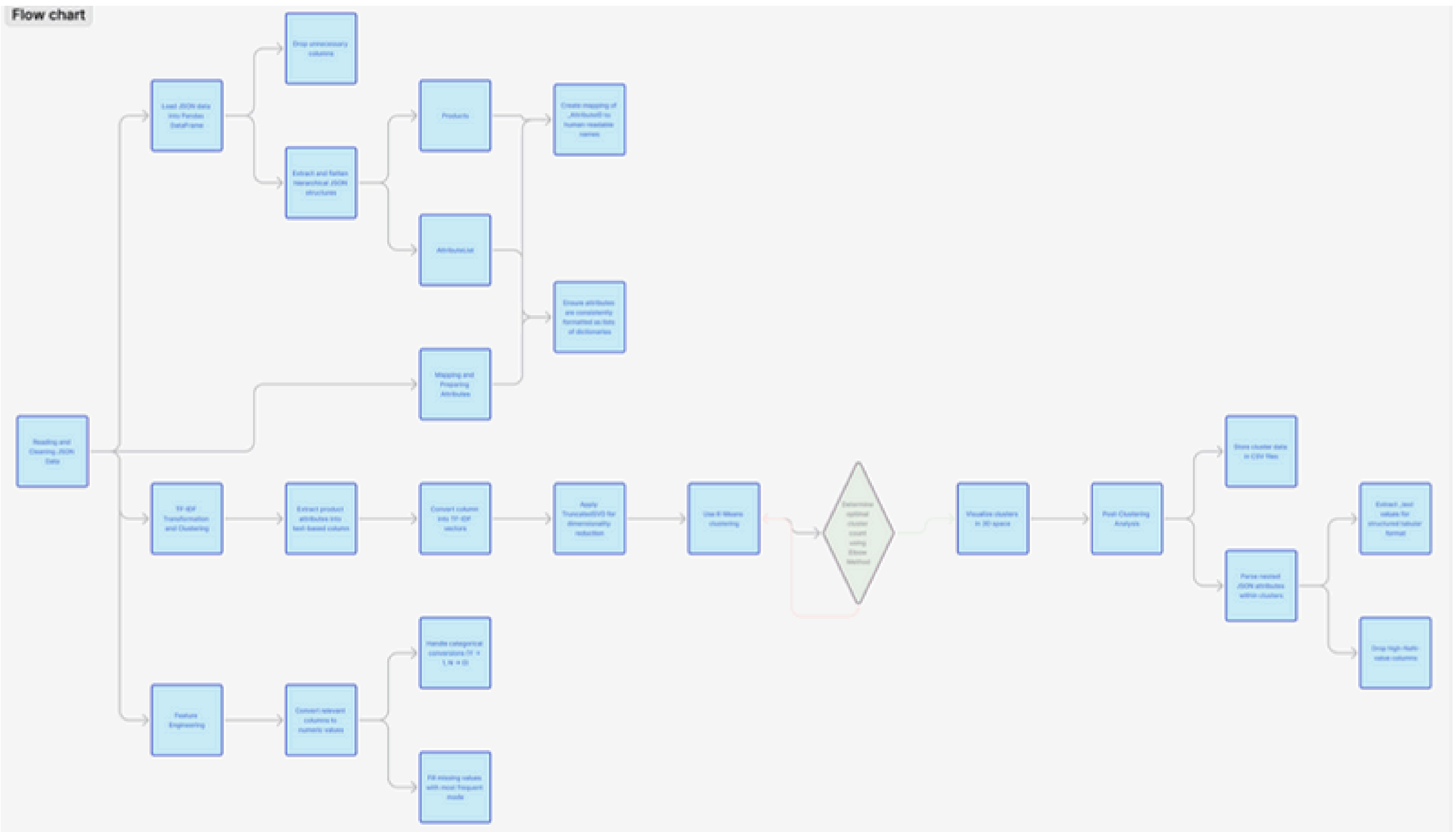
CHATBOT

COMPETITIVE BENCHMARKING

	PROS	PAIN POINTS	
Informatica MDM	**Informatica is a well-established player in the MDM market. Its MDM solution offers comprehensive data management capabilities, scalability, and support for various data domains.**	**: Depending on specific use cases, pricing and implementation complexity can be factors to consider.**	
IBM InfoSphere MDM	• IBM provides a robust MDM solution with a focus on data governance and integration. It supports multiple domains and offers scalability for large enterprises.	IBM solutions might be perceived as complex, and implementation may require skilled resources.	
SAP Master Data Governance	SAP's MDM solution is tightly integrated with its broader suite of enterprise applications. It provides strong governance features and supports various data domains.	Implementation can be complex, and it's most suitable for organizations already invested in the SAP ecosystem.	
Reltio	Reltio is known for its cloud-based MDM platform. It focuses on delivering a connected customer experience and real-time data management.	It may be more suitable for organizations preferring cloud-native solutions.	

STEPS

1. **READING AND CLEANING JSON DATA**
   - You load JSON data into a Pandas DataFrame.
   - Drop unnecessary columns like metadata fields.
   - Extract and flatten hierarchical JSON structures from Products and AttributeList.
2. **MAPPING AND PREPARING ATTRIBUTES**
   - You create a mapping of _AttributeID to human-readable names.
   - Ensure the extracted attributes are consistently formatted as lists of dictionaries.
3. **TF-IDF TRANSFORMATION AND CLUSTERING**
   - You extract product attributes into a single text-based column.
   - Convert this column into TF-IDF vectors.
   - Apply TruncatedSVD for dimensionality reduction.
   - Use K-Means clustering and determine the optimal cluster count using the Elbow Method.
   - Visualize clusters in 3D space.
4. **POST-CLUSTERING ANALYSIS**
   - Store cluster data in CSV files.
   - Parse nested JSON attributes within clusters.
   - Extract _text values from attributes for a structured tabular format.
   - Drop high-NaN-value columns.
5. **FEATURE ENGINEERING**
   - Convert relevant columns to numeric values.
   - Handle categorical conversions (Y → 1, N → 0).
   - Fill missing values with the most frequent mode.

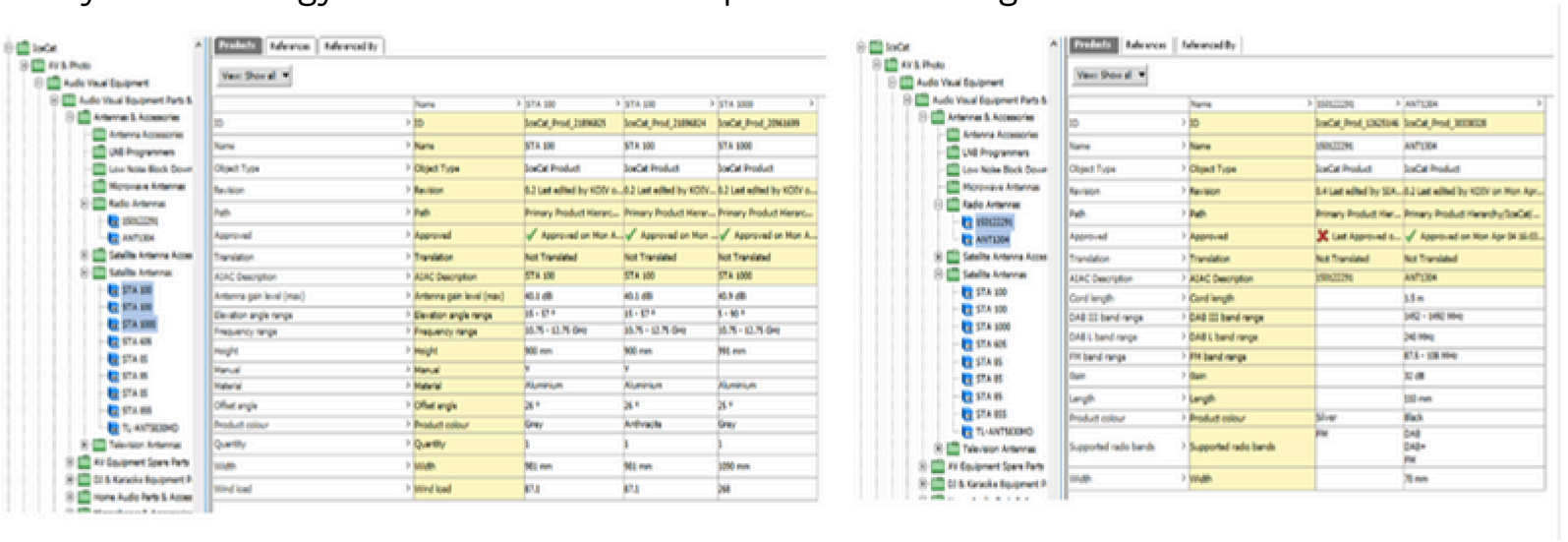

Flow chart
Drop unnecessary columns
Load JSON data into Pandas DataFrame
Extract and flatten hierarchical JSON structures
Products
Create mapping of _Attributes to human-readable names
Attributes.txt
Ensure attributes are consistently formatted as lists of dictionaries
Mapping and Preparing Attributes
Reading and Cleaning JSON Data
TF-IDF Transformation and Clustering
Extract product attributes into text-based column
Convert column into TF-IDF vectors
Apply TruncatedSVD for dimensionality reduction
Use K-Means clustering
Determine optimal cluster count using Elbow Method
Visualize clusters in 3D space
Post-Clustering Analysis
Store cluster data in CSV files
Parse nested JSON attributes within clusters
Extract Json values for structured tabular format
Drop high-Nulls value columns
Feature Engineering
Convert relevant columns to numeric values
Handle categorical conversions (1:1, 1:N, n:n)
Fill missing values with most frequent mode

Project Overview

Data maintenance is a tedious process. In order to reduce it, products have been grouped into families, so that common features such as inheritance can be controlled. For example, several T-shirts in various colors and sizes are grouped together because they are composed of the same fabric.

Goal: Create logical product families by grouping related products from a sizable dataset.

The primary methodology to achieve this is unsupervised learning.



Why Clustering?

The ability to find significant patterns and insights in the data that can guide strategic decision-making and propel corporate success is what makes clustering so important to a corporation.

Market Segmentation: Based on behavioral, demographic, or preference similarities, clustering aids in the division of consumers or market segments.

Product Development: Businesses can find product categories or traits that appeal to particular client segments by grouping products according to features, usage trends, or customer feedback.

Resource Allocation: By pinpointing regions of concentration or investment that have the highest potential for success, clustering can help in resource allocation optimization.

Dataset Overview

As can be seen below, the dataset has two major rows that provide us with grouping information (Products and Attributes).

The nested JSON in both rows contains data for different products and attributes.

The dataset, which will serve as the foundation for our clustering, contains over 4,000 attributes and approximately 12,000 goods overall.

In [2]:
```
import pandas as pd
raw_data=pd.read_json('convertjson.json')
from pandas import json_normalize
# raw_data['AttributeList']
raw_data
```

Out[2]:

	STEP-ProductInformation
AttributeList	{'Attribute': [{'Name': 'Screwdriver bit', 'Li...
ListOfValuesGroupList	{'ListOfValuesGroup': [{'Name': 'Lists of Valu...
ListsOfValues	{'ListOfValue': [{'Name': 'Seals material', 'V...
Products	{'Product': [{'Name': 'LR2000 2000W 230V Power...
_ContextID	Context1
_ExportContext	Context1
_ExportTime	2024-01-08 13:32:56
_UseContextLocale	false
_WorkspaceID	Main

Cleaning this, we get

```
df = pd.DataFrame(raw_data)
df=df.drop(['ListOfValuesGroupList', 'ListsOfValues','_ContextID','_Export
df
```

	STEP-ProductInformation
AttributeList	{'Attribute': [{'Name': 'Screwdriver bit', 'Li...
Products	{'Product': [{'Name': 'LR2000 2000W 230V Power...

Clustering Based on Attributes

STEP 1 – FLATTENING JSON

PREPROCESSING OF DATA

Obtaining product information in a more understandable way by flattening the hierarchical JSON inside the dictionary.

	Name	_ID	_UserTypeID	Values.Value	Values.MultiValue	ClassificationCrossReference	ClassificationReference
0	LR2000 2000W 230V Power Conditioner with Autom...	IceCat_Prod_1668262	IceCat_Product	[{'_AttributeID': 'IceCat_Attrib_22658', '_Uni...	[{'Value': ['Y'], '_AttributeID': 'IceCat_Attr...	NaN	NaN
1	1200W 120V Power Conditioner with Automatic Vo...	IceCat_Prod_1668268	IceCat_Product	[{'_AttributeID': 'IceCat_Attrib_22658', '_Uni...	[{'Value': ['Plastic'], '_AttributeID': 'IceCa...	NaN	NaN
2	600W 120V Power Conditioner with Automatic Vol...	IceCat_Prod_2273759	IceCat_Product	[{'_AttributeID': 'IceCat_Attrib_22658', '_Uni...	[{'Value': ['Acrylonitrile butadiene styrene (...	NaN	NaN
3	Isobar 2-Outlet Low-Profile Professional Audio...	IceCat_Prod_6291127	IceCat_Product	[{'_AttributeID': 'IceCat_Attrib_22658', '_Uni...	[{'Value': ['China'], '_AttributeID': 'IceCat_...	NaN	NaN
4	2400W 120V Power Conditioner with Automatic Vo...	IceCat_Prod_2273755	IceCat_Product	[{'_AttributeID': 'IceCat_Attrib_22658', '_Uni...	[{'Value': ['Y'], '_AttributeID': 'IceCat_Attr...	NaN	NaN
...	...	...	...	...	...	...	...
2181	Fugen Neu Express	IceCat_Prod_55183541	IceCat_Product	[{'_AttributeID': 'IceCat_Attrib_1675', '_tex...	[{'Value': ['Bathroom', 'Kitchen'], '_Attribut...	NaN	NaN
2182	Fliesen Fix	IceCat_Prod_55183495	IceCat_Product	[{'_AttributeID': 'IceCat_Attrib_1675', '_tex...	[{'Value': ['Bathroom'], '_AttributeID': 'IceC...	NaN	NaN

STEP 2 – ATTRIBUTES

PREPROCESSING OF DATA

Flattening attributes in the same way and creating a mapping of attribute IDs with attribute names in a separate dictionary.

	Name	UserTypeLink	_ID	_MultiValued	_ProductMode	_FullTextIndexed	_ExternallyMaintained	_Derived	_HierarchicalFiltering
0	Screwdriver bit	[{'_UserTypeID': 'IceCat_Product'}]	IceCat_Attrib_6786	false	Normal	false	false	false	false
1	Chemical flux-resistant	[{'_UserTypeID': 'IceCat_Product'}]	IceCat_Attrib_40910	false	Normal	false	false	false	false
2	Impact energy (max)	[{'_UserTypeID': 'IceCat_Product'}]	IceCat_Attrib_6782	false	Normal	false	false	false	NaN
3	Idle speed (1st gear)	[{'_UserTypeID': 'IceCat_Product'}]	IceCat_Attrib_6787	false	Normal	false	false	false	NaN
4	Handguard	[{'_UserTypeID': 'IceCat_Product'}]	IceCat_Attrib_27423	false	Normal	false	false	false	false
...	...	...	...	...	...	...	...	...	...
4581	Colour name	[{'_UserTypeID': 'IceCat_Product'}]	IceCat_Attrib_17956	false	Normal	false	false	false	NaN
4582	AC input frequency	[{'_UserTypeID': 'IceCat_Product'}]	IceCat_Attrib_8484	false	Normal	false	false	false	NaN
4583	Number of frames	[{'_UserTypeID': 'IceCat_Product'}]	IceCat_Attrib_9170	false	Normal	false	false	false	NaN
4584	Number of accessories included	[{'_UserTypeID': 'IceCat_Product'}]	IceCat_Attrib_29632	false	Normal	false	false	false	NaN
4585	AC input voltage	[{'_UserTypeID': 'IceCat_Product'}]	IceCat_Attrib_8491	false	Normal	false	false	false	NaN

4586 rows × 27 columns

Mapping Attribute IDs to corresponding Name

	ID	Name
0	IceCat_Attrib_6786	Screwdriver bit
1	IceCat_Attrib_40910	Chemical flux-resistant
2	IceCat_Attrib_6782	Impact energy (max)
3	IceCat_Attrib_6787	Idle speed (1st gear)
4	IceCat_Attrib_27423	Handguard
...	...	...
4581	IceCat_Attrib_17956	Colour name
4582	IceCat_Attrib_8484	AC input frequency
4583	IceCat_Attrib_9170	Number of frames
4584	IceCat_Attrib_29632	Number of accessories included
4585	IceCat_Attrib_8491	AC input voltage

4586 rows × 2 columns

STEP 3 – REMOVING NAN, COMBINING VALUES

PREPROCESSING OF DATA

Removed NaN rows and subsequently combining Values.Values and Values.MultiValues column’s into a single Attribute’s column

Values.Value	Values.MultiValue	Attributes
[{'_AttributeID': 'IceCat_Attrib_22658', '_Uni...	[{'Value': ['Y'], '_AttributeID': 'IceCat_Attr...	[{'_AttributeID': 'IceCat_Attrib_22658', '_Uni...
[{'_AttributeID': 'IceCat_Attrib_22658', '_Uni...	[{'Value': ['Plastic'], '_AttributeID': 'IceCa...	[{'_AttributeID': 'IceCat_Attrib_22658', '_Uni...
[{'_AttributeID': 'IceCat_Attrib_22658', '_Uni...	[{'Value': ['Acrylonitrile butadiene styrene (...	[{'_AttributeID': 'IceCat_Attrib_22658', '_Uni...
[{'_AttributeID': 'IceCat_Attrib_22658', '_Uni...	[{'Value': ['China'], '_AttributeID': 'IceCat_...	[{'_AttributeID': 'IceCat_Attrib_22658', '_Uni...
[{'_AttributeID': 'IceCat_Attrib_22658', '_Uni...	[{'Value': ['Y'], '_AttributeID': 'IceCat_Attr...	[{'_AttributeID': 'IceCat_Attrib_22658', '_Uni...
...	...	...
[{'_AttributeID': 'IceCat_Attrib_1675', '__tex...	[{'Value': ['Bathroom'], '_AttributeID': 'IceC...	[{'_AttributeID': 'IceCat_Attrib_1675', '__tex...
[{'_AttributeID': 'IceCat_Attrib_13591', '__te...	[{'Value': ['Bathroom', 'Kitchen'], '_Attribut...	[{'_AttributeID': 'IceCat_Attrib_13591', '__te...
[{'_AttributeID': 'IceCat_Attrib_1675', '__tex...	[{'Value': ['Bathroom', 'Kitchen'], '_Attribut...	[{'_AttributeID': 'IceCat_Attrib_1675', '__tex...
[{'_AttributeID': 'IceCat_Attrib_1675', '__tex...	[{'Value': ['Bathroom'], '_AttributeID': 'IceC...	[{'_AttributeID': 'IceCat_Attrib_1675', '__tex...
[{'_AttributeID': 'IceCat_Attrib_13591', '__te...	[{'Value': ['Ready mixed'], '_AttributeID': 'I...	[{'_AttributeID': 'IceCat_Attrib_13591', '__te...

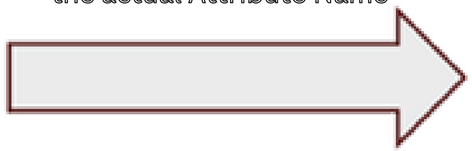
STEP 4 – ATTRIBUTE NAMES

PREPROCESSING OF DATA

Each product attribute now has an associated attribute ID, and each row in the attribute column is a list of dictionaries representing these attributes. To facilitate data grouping, we have mapped these IDs to their corresponding attribute names, as the names are crucial for this purpose. Consequently, we replace the IDs with their respective names, as the IDs themselves are not very informative.

```
 '__text': '444.5'},
{'_AttributeID': 'IceCat_Attrib_19016',
 '_UnitID': 'IceCat_Unit_38',
 '__text': '24990'},
{'_AttributeID': 'IceCat_Attrib_762',
 '_UnitID': 'IceCat_Unit_38',
 '__text': '6030'},
{'_AttributeID': 'IceCat_Attrib_16240',
 '_UnitID': 'IceCat_Unit_81',
 '__text': '1200'},
{'_AttributeID': 'IceCat_Attrib_23056', '__text': '10037332040289'},
{'_AttributeID': 'IceCat_Attrib_3808',
 '_UnitID': 'IceCat_Unit_24',
 '__text': '215.9'},
{'_AttributeID': 'IceCat_Attrib_3806',
 '_UnitID': 'IceCat_Unit_24',
 '__text': '259.1'},
{'_AttributeID': 'IceCat_Attrib_5932',
 '_UnitID': 'IceCat_Unit_44',
 '__text': '2000'},
{'_AttributeID': 'IceCat_Attrib_2556',
 '_UnitID': 'IceCat_Unit_384',
 '__text': '6'},
{'_AttributeID': 'IceCat_Attrib_3807',
 '_UnitID': 'IceCat_Unit_24',
 '__text': '203.2'},
```

Replace the IceCat Attribute with the actual Attribute Name



```
[{'_AttributeID': 'Master (outer) case length',
  '_UnitID': 'IceCat_Unit_24',
  '__text': '444.5'},
 {'_AttributeID': 'Master (outer) case gross weight',
  '_UnitID': 'IceCat_Unit_38',
  '__text': '24990'},
 {'_AttributeID': 'Package weight',
  '_UnitID': 'IceCat_Unit_38',
  '__text': '6030'},
 {'_AttributeID': 'UPS AC suppression rating',
  '_UnitID': 'IceCat_Unit_81',
  '__text': '1200'},
 {'_AttributeID': 'Master (outer) case GTIN (EAN/UPC)',
  '__text': '10037332040289'},
 {'_AttributeID': 'Package width',
  '_UnitID': 'IceCat_Unit_24',
  '__text': '215.9'},
 {'_AttributeID': 'Package depth',
  '_UnitID': 'IceCat_Unit_24',
  '__text': '259.1'},
 {'_AttributeID': 'Output power',
  '_UnitID': 'IceCat_Unit_44',
  '__text': '2000'},
 {'_AttributeID': 'AC outlets quantity',
  '_UnitID': 'IceCat_Unit_384',
  '__text': '6'},
```

OUTCOMES

This basic preprocessing step has laid the foundation for our clustering analysis by enabling us to construct the fundamental building blocks. An important observation in the dataset is that it contains a diverse range of products, making two randomly chosen products potentially very different in terms of their attributes.

For example, a chair may have attributes related to legs, while a pillow would not.

Based on this observation, I decided to cluster the products according to their existing attributes, specifically the names of these attributes. After forming initial clusters based on attribute names, we can further refine our clustering by subdividing these clusters to obtain more similar groups.

MORE PREPROCESSING

First, we extract the names of attributes from the dictionaries in the attributes column for each product and store them as lists in a new column named 'final column'.

Next, we concatenate the items in these lists to convert them into strings and add them to another new column. This results in a column named 'tf-idf column', where each row contains a string.

With the 'tf-idf column' in place, I will apply clustering on it. However, before proceeding with clustering, I need to convert the string data into numerical data using the **TF-IDF (Term Frequency-Inverse Document Frequency) technique**.

Attributes	final_column	tf-idf column
[{'_AttributeID': 'Master (outer) case length'...	[Master (outer) case length, Master (outer) ca...	Master (outer) case length Master (outer) case...
[{'_AttributeID': 'Master (outer) case length'...	[Master (outer) case length, Master (outer) ca...	Master (outer) case length Master (outer) case...
[{'_AttributeID': 'Master (outer) case length'...	[Master (outer) case length, Master (outer) ca...	Master (outer) case length Master (outer) case...
[{'_AttributeID': 'Master (outer) case length'...	[Master (outer) case length, Master (outer) ca...	Master (outer) case length Master (outer) case...
[{'_AttributeID': 'Master (outer) case length'...	[Master (outer) case length, Master (outer) ca...	Master (outer) case length Master (outer) case...
...	...	...
'_AttributeID': 'Quantity', '__text': '1'}, ...	[Quantity, Drying time, Suitable for indoor us...	Quantity Drying time Suitable for indoor use S...

WHY TF-IDF?

Dimensionality Reduction: By using TF-IDF, common and less important terms are down-weighted while unique and relevant terms are up-weighted. This helps reduce the dimensionality of the data and focus on the distinguishing features of each document, which is beneficial for clustering.

Normalization: TF-IDF normalizes the term frequencies based on their occurrence across the entire dataset. This ensures that the clustering algorithm is not biased towards terms that are simply more frequent overall, but rather towards terms that are relatively more important within each document.

Relevance of Terms: TF-IDF reflects the importance of terms in a document relative to a corpus. In the context of clustering based on textual data, it helps identify the significance of terms within each document (or, in this case, each product attribute list).

Compatibility with Clustering Algorithms: Many clustering algorithms, such as K-means, work well with numerical data. TF-IDF provides a numerical representation of textual data that is suitable for clustering algorithms.

THIS IS WHY - OUTCOMES

Win 1: String data converted to numerical data to which clustering can be applied

Win 2: By using SVD, we observe reduced dimensions to 100 columns. In general, to address this issue of dimension of the resulting TF-IDF matrix being typically large, and to reduce computational complexity, dimensionality reduction techniques like SVD or PCA are used.

```
(0, 165)      0.022296662615797 82
(0, 600)      0.023351023527290186
(0, 470)      0.05414396833253917
(0, 377)      0.07767200411057809
(0, 135)      0.08223092335900567
(0, 754)      0.08223092335900567
(0, 533)      0.093795051963603 73
(0, 514)      0.032599977078239674
(0, 204)      0.093795051963603 73
(0, 398)      0.08914458692805144
(0, 440)      0.08876467895682678
(0, 220)      0.020894362786532675
```

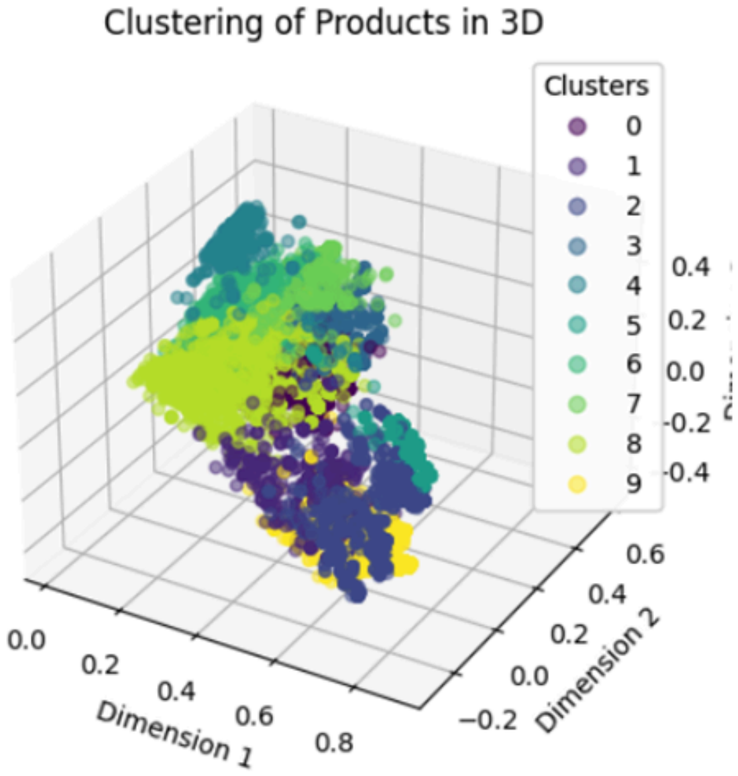
CLUSTERING WITH K-MEANS

After reducing the dimensions, I used K-Means clustering to group the data.

Why K-Means: K-Means is computationally efficient and can handle large datasets with a significant number of samples. It works well with numerical data and is relatively easy to implement. Additionally, methods like TF-IDF integrate well with K-Means.

Finding the Appropriate Number of Clusters:

Determining the appropriate number of clusters can be achieved using methods like the elbow method, which is a common approach in clustering analysis. By setting an optimal value of K (which can be subjective based on the requirement), I obtained the following clusters.



POTENTIAL ENHANCEMENTS

✅ I**mprove NaN Handling:** Instead of dropping columns outright, consider imputation techniques (mean, median, mode).

✅ **Optimize K-Means:** Try DBSCAN or Agglomerative Clustering, which can handle varying densities.

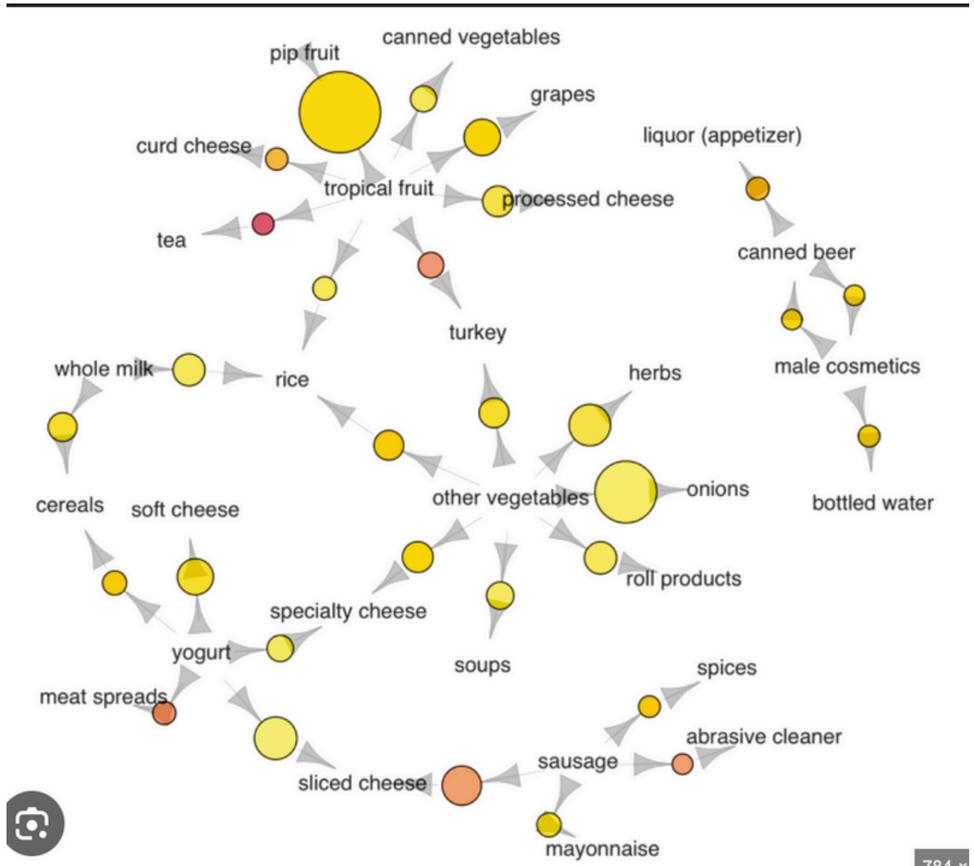
✅ **Feature Scaling:** Before clustering, normalize numerical features using MinMaxScaler or StandardScaler for better performance.

✅ **Parallel Processing:** Large datasets? Use Dask or Modin instead of Pandas for efficiency.

APRIOPRI VS CLUSTERING

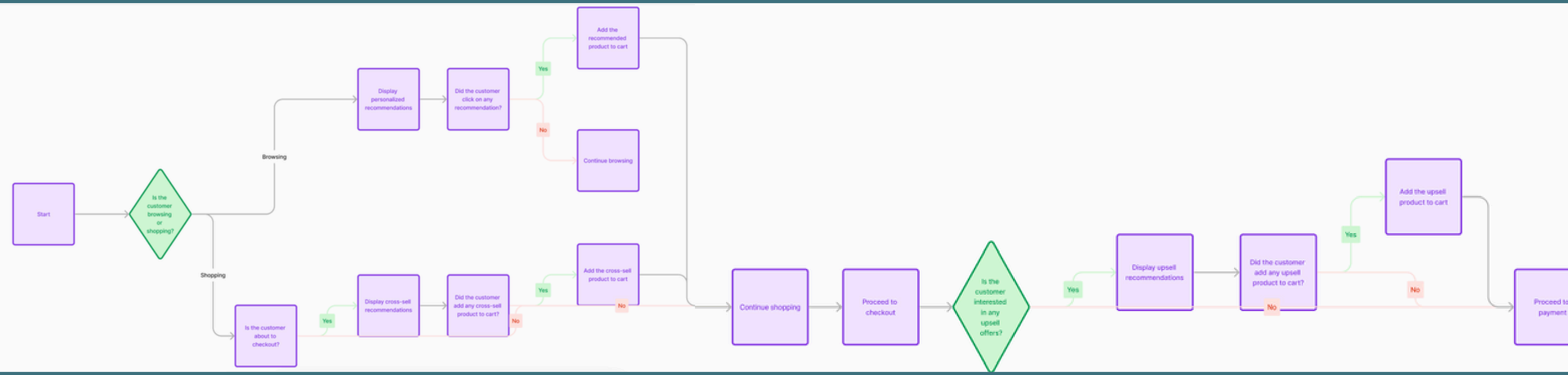
Feature	TF-IDF + K-Means + SVD	Apriori
Type of Data	Textual & attribute-based data	Transactional (market basket) data
Handling High-Dimensional Data	Efficient (SVD reduces dimensions)	Poor (computationally expensive)
Scalability	Fast & scalable (O(nk) complexity)	Slow (exponential complexity)
Cold-Start Problem	Works without prior interactions	Requires transaction history
Understanding Similarities	Captures semantic similarity (TF-IDF)	Only finds co-occurrence patterns
Recommendation Type	Cluster-based recommendations	Rule-based ("If X, then Y")
Computational Efficiency	Fast (vectorized operations)	Slow (frequent itemset mining)
Use Case	Attribute-based recommendations	Market basket analysis
Interpretability	Moderate (clusters need labeling)	High (clear association rules)
Best For	Content-based recommendations	Finding frequently bought item sets

TECHNICAL ANALYSIS

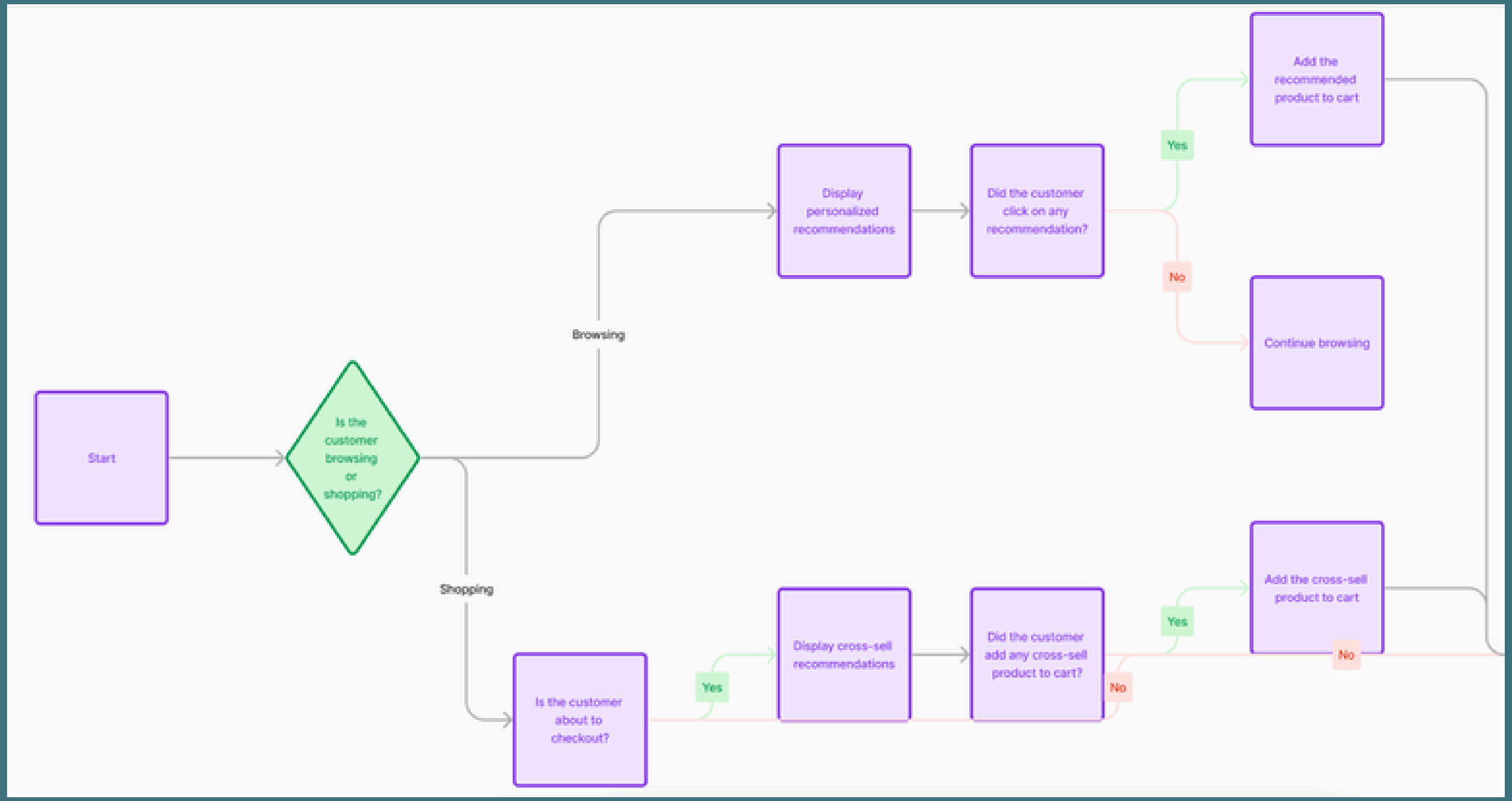


BUSINESS MODEL

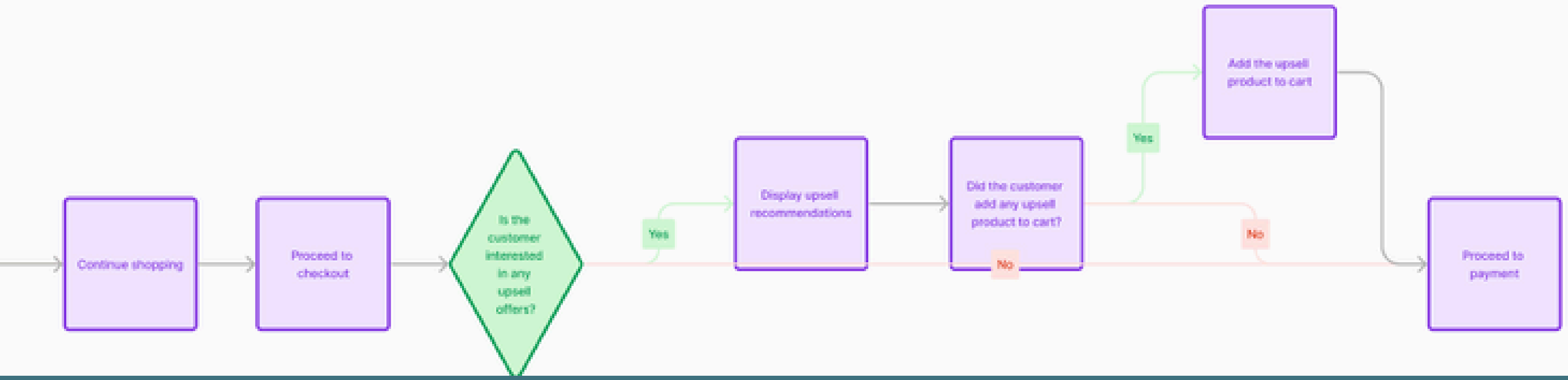
FLOWCHART


Start
Is the customer browsing or shopping?
Browsing
Shopping
Display personalized recommendations
Did the customer click on any recommendation?
Yes
No
Add the recommended product to cart
Continue browsing
Is the customer about to checkout?
Yes
Display cross-sell recommendations
Did the customer add any cross-sell product to cart?
Yes
No
Add the cross-sell product to cart
No
Continue shopping
Proceed to checkout
Is the customer interested in any upsell offers?
Yes
Display upsell recommendations
Did the customer add any upsell product to cart?
Yes
No
Add the upsell product to cart
No
Proceed to payment

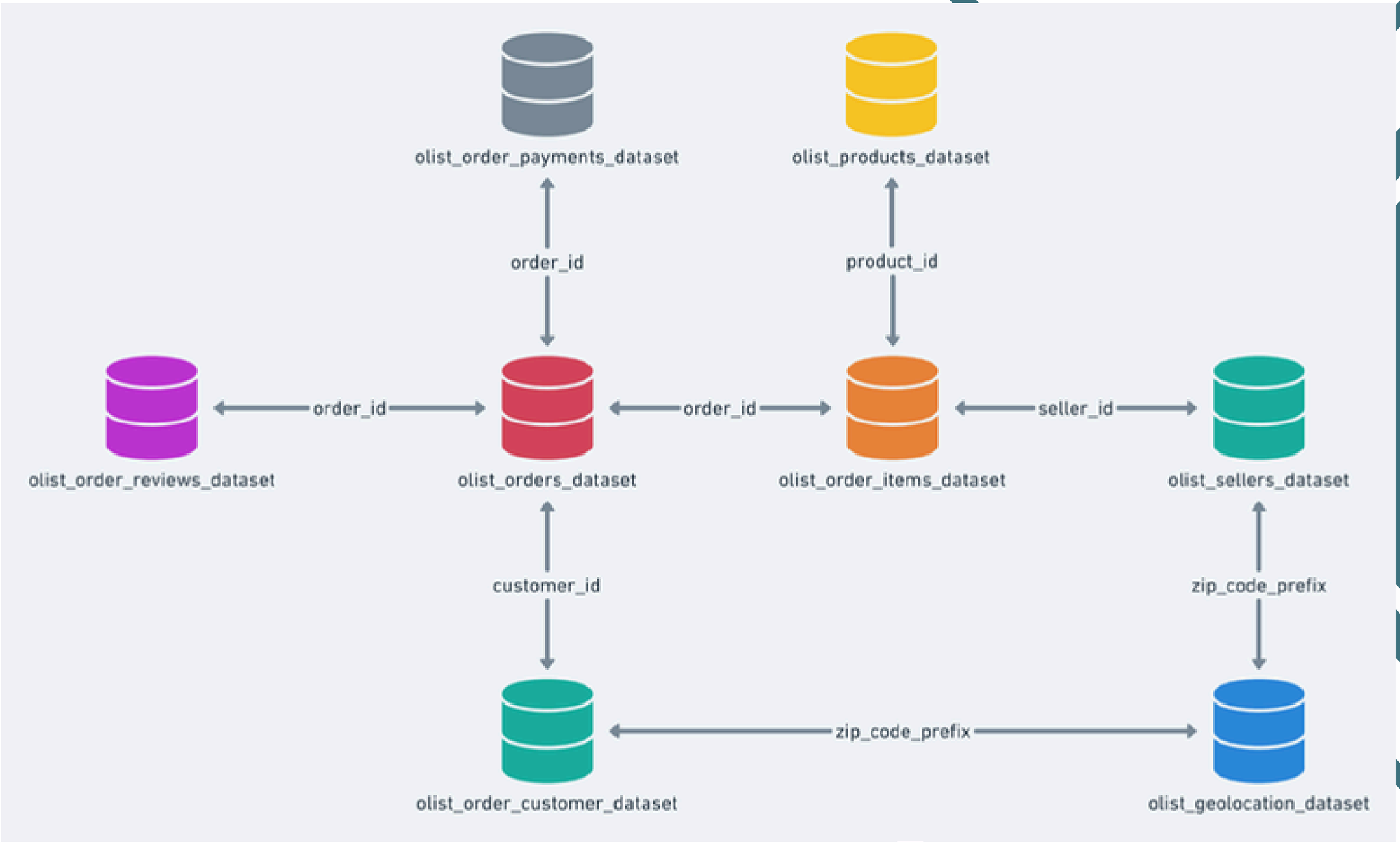
FLOWCHART


Start
Is the customer browsing or shopping?
Browsing
Shopping
Display personalized recommendations
Did the customer click on any recommendation?
Yes
No
Add the recommended product to cart
Continue browsing
Is the customer about to checkout?
Yes
Display cross-sell recommendations
Did the customer add any cross-sell product to cart?
Yes
No
Add the cross-sell product to cart
No

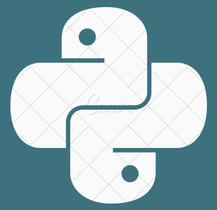
FLOWCHART


Continue shopping
Proceed to checkout
Is the customer interested in any upsell offers?
Yes
Display upsell recommendations
Did the customer add any upsell product to cart?
Yes
Add the upsell product to cart
No
No
Proceed to payment

HOW THE DATA IS CONNECTED



TECH STACK



PYTHON

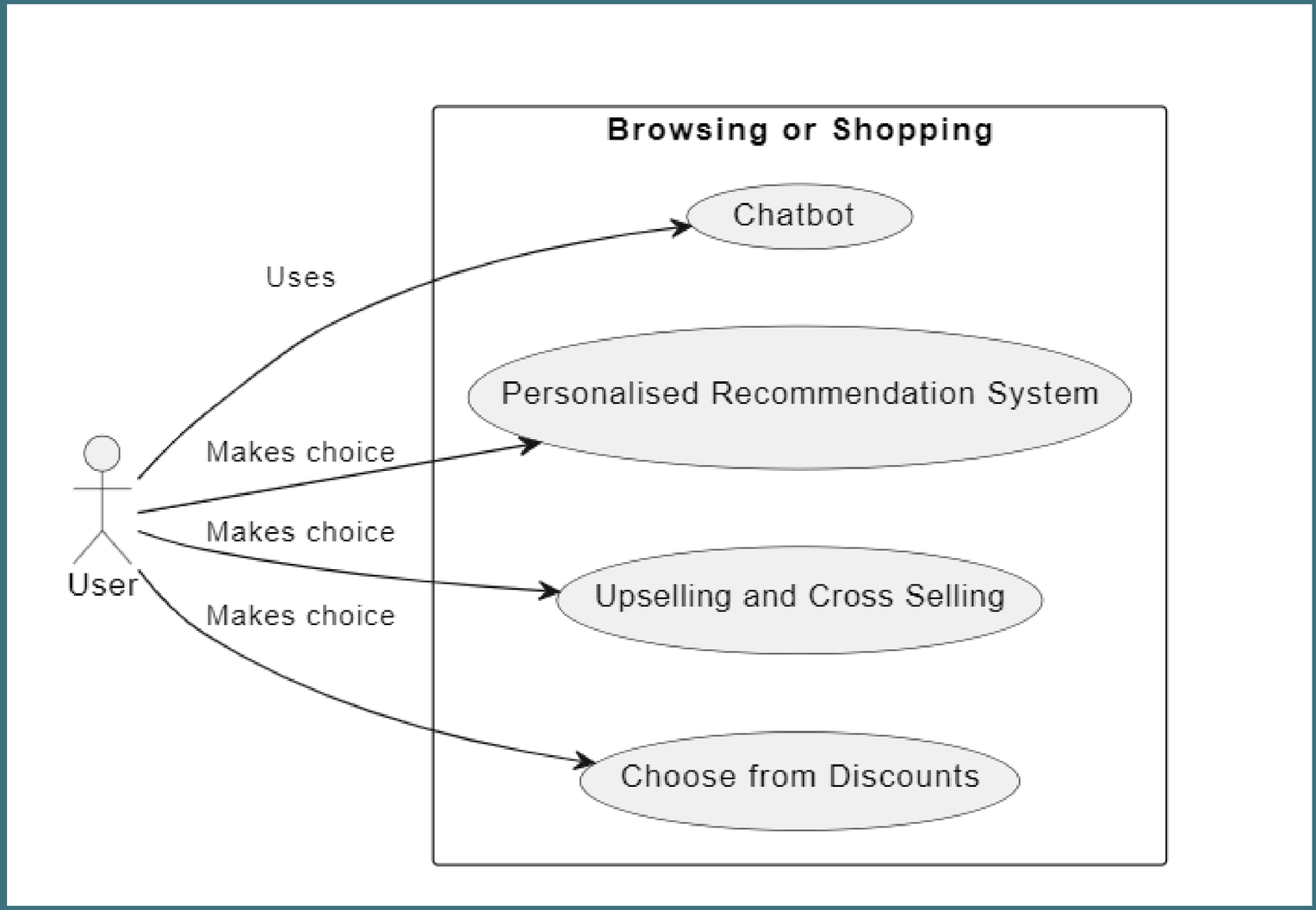
CUSTOM CHATBOT

DOCKER AND KUBERNETES

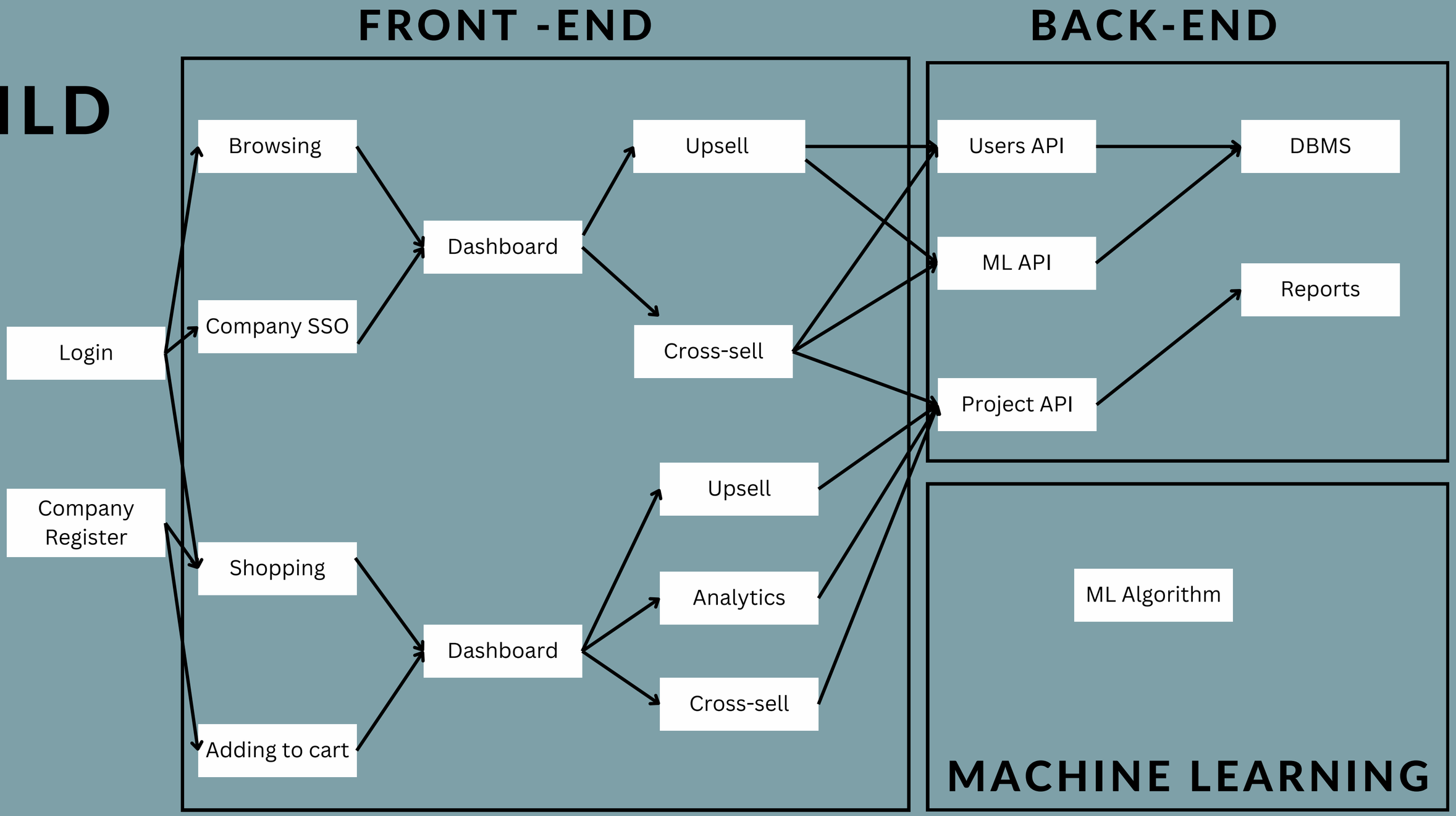
FLASK OR DJANGO

API

USE CASE


Browsing or Shopping
Chatbot
Uses
Personalised Recommendation System
Makes choice
Makes choice
User
Upselling and Cross Selling
Makes choice
Choose from Discounts

HLD


FRONT -END
BACK-END
Browsing
Company SSO
Login
Company Register
Shopping
Adding to cart
Dashboard
Dashboard
Upsell
Cross-sell
Upsell
Analytics
Cross-sell
Users API
ML API
Project API
DBMS
Reports
ML Algorithm
MACHINE LEARNING

Technical Implementation

1

DATA PREPROCESSING

XML viewing in tree structure and parsing

- *View in treeL firstobject*
- *Parse Libraries to parse: xml.etree.ElementTree, lxml and xmltodict*

2

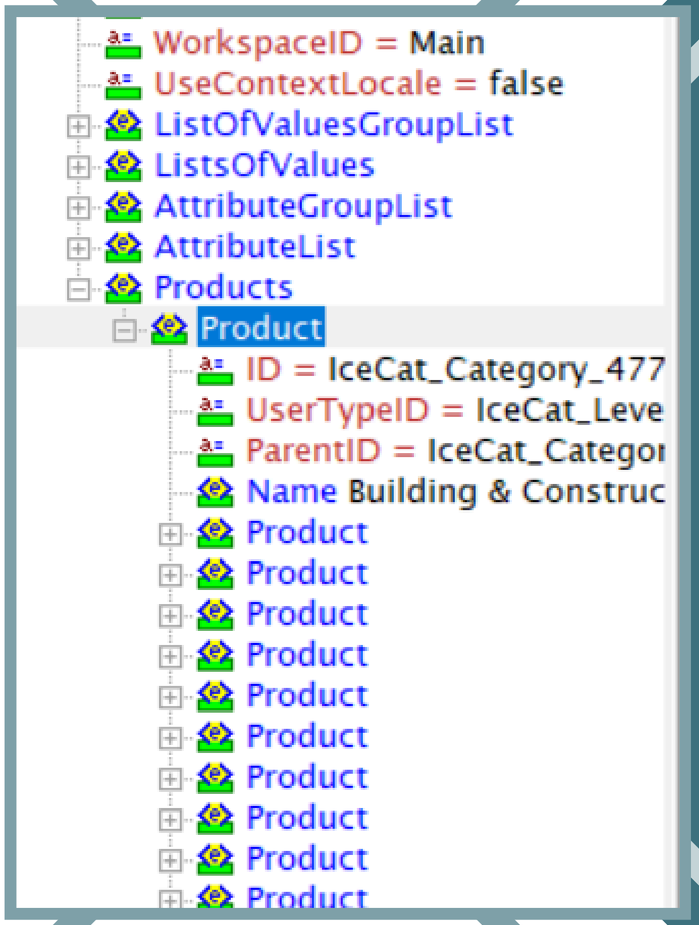
DATA EXPLORATION

Libraries: Pandas, NumPy, Seaborn

3

ASSOCIATION RULE MINING

Libraries: mlxtend for Apriopri



Technical Implementation

1

FILTERING RULE

Filtering rules based on support, confidence or lift thresholds

2

INTEGRATION WITH BUSINESS LOGIC

Flask: Integrate recommendation engine into a web application

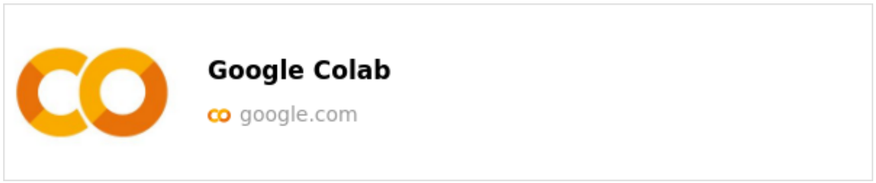
3

SCALABILITY AND DEPLOYMENT

MongoDB for scalability

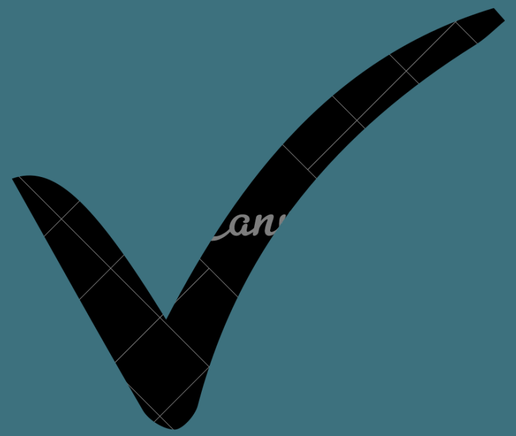
	antecedents	consequents	antecedent support	consequent support	support	confidence
8	(MILK)	(BREAD)	0.25	0.65	0.20	0.80
18	(MAGGI)	(TEA)	0.25	0.35	0.20	0.80
9	(SUGER)	(BREAD)	0.30	0.65	0.20	0.67
13	(CORNFLAKES)	(COFFEE)	0.30	0.40	0.20	0.67
14	(SUGER)	(COFFEE)	0.30	0.40	0.20	0.67
0	(BISCUIT)	(BREAD)	0.35	0.65	0.20	0.57
10	(TEA)	(BREAD)	0.35	0.65	0.20	0.57
19	(TEA)	(MAGGI)	0.35	0.25	0.20	0.57

DEMO

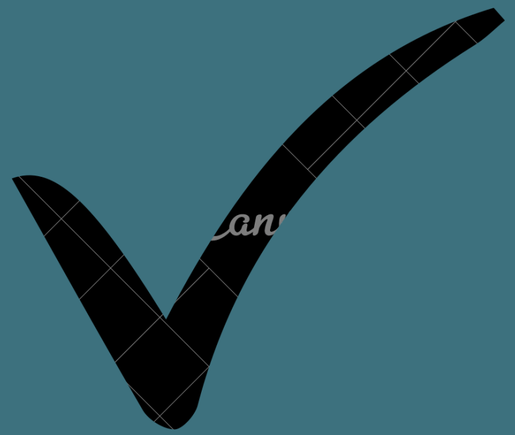


CHALLENGES

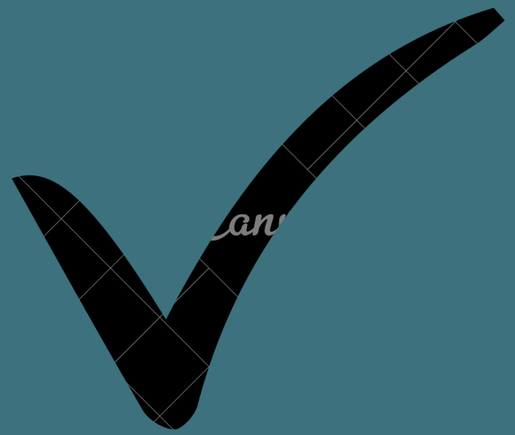
UPSELL



CROSS-SELL



PERSONALISED RECOMMENDATION



CONCLUSIONS

SCALABILITY

As the dataset grows, the complexity of the Apriori algorithm may lead to performance issues.

RULE INTERPRETABILITY

Interpreting and explaining complex association rules to stakeholders can be challenging.

COLD START PROBLEM

Difficulty in providing recommendations for new or less-purchased products (cold start problem).