UNIVERSITI MALAYA

WIE3007

DATA MINING AND WAREHOUSING

SEMESTER 1 SESSION 2023/2024

GROUP ASSIGNMENT

TITLE: COFFEE SHOP PRODUCT AND CUSTOMER ANALYSIS

LECTURER NAME: ASSOC. PROF. DR. TEH YING WAH

Github Link: zrlxkai/DMGroupProject (github.com)

YouTube Video Presentation Link: https://youtu.be/I_ox30Zt9nw

Group Members	**Matric Number**
Amrin Hafiz bin Eddy Rosyadie	17205136
Azrul Haikal bin Uhaidi	17201828
Muhammad Rafid Ikhwan Bin Samsuri	17204331
Nurul Filzah binti Abdul Hadi	17205112

Wan Suraya binti Wan Mohd Lotfi	U2005345

Table of Contents

1.0 Introduction

1.1 Problem Statement

1) Understanding Customer Behavior

Understanding how customers behave is an important and vital issue for many companies, particularly in the big data and digital transformation age. Understanding what consumers need, desire, prefer, and anticipate from a product or service as well as how they engage with it is made easier with the use of customer behavior analysis. According to a McKinsey analysis, the US retail industry may use big data and analytics to understand and impact consumer behavior, leading to an increase in revenue of $1.7 trillion annually by 2025. In order to handle the complexity and variety of customer behavior data and produce forecasts and actionable insights for organizations, research on customer behavior requires complex analytical techniques and tools.

2) Assessing Product Popularity and Demand

The technique of projecting future demand for a good or service using market trends, historical data, and other considerations is known as demand forecasting. Businesses may maximize their inventory levels, cut expenses, boost sales, and enhance customer happiness with the aid of accurate demand forecasting. According to a research by the Institute of Business Forecasting and Planning, consumer products businesses' average prediction accuracy was just 77%, which means that the actual demand differed from the predicted demand 23% of the time. Demand patterns can fluctuate based on a number of variables, including seasonality, competition, customer preferences, price adjustments, promotions, and unforeseen events, making demand forecasting a difficult and demanding activity.

3) Uncovering Seasonal Buying Trends

Seasonal demand forecasting is one of the most important tasks for eCommerce companies, according to research by WareIQ, as it allows them to take advantage of times when demand is higher, avoid having too much or too little stock, and choose products wisely. However, variables including short product life cycles, seasonal changes, a lack of historical data, and hidden seasonality might make it difficult to estimate seasonal demand. Such data can assist buyers and realtors in planning ahead and making educated selections.

1.2 Business Objectives

1) To analyze customer behavior according to different generations in order to customize marketing strategies and product offerings that effectively target certain age groups.
2) To determine coffee purchasing patterns on a daily basis in order to distribute resources effectively, synchronize promotions with high-demand periods, and improve operational efficiency.
3) To investigate the popularity of products within each generation in order to optimize the product selection, enhance inventory control, and accommodate the various tastes of customers across different age segments.
4) To analyze what generation is the particular customer in based on their customer behavior pattern

1.3 Dataset Description

This sample data module includes standard retail information from a hypothetical coffee shop. The data source is contained in an uploaded file with the name April Sales.zip. Source: IBM. A dataset has been created for a fictional franchise of three New York City coffee shops. In order to identify factors that have contributed to their success and, eventually, to make data-driven decisions, the franchise purchased IBM Cognos Analytics, and the range of data is during April 2019.

The people in charge of the coffee franchise, Amber and Sandeep, started by uploading their data using a series of spreadsheets, and then creating a dataset. They developed a marketing dashboard and an operations dashboard using this dataset. Attributes included in this dataset are as follows:

1) Sales Receipt

Attributes	**Description**
sales_id	Unique ID for every sales
transaction_id	Unique ID for every transaction
transaction_date	Date of transaction
transaction_time	Time of transaction
customer_id	Unique ID for every customer
instore_yn	Method of buying i.e. instore or online
product_id	Unique ID for every product
quantity	Quantity purchased for each transaction
unit_price	Price for every unit of product

Table 1.1: Attributes for 'Sales Receipt' data

2) Customer

customer_id	Unique ID for every customer
customer_firstName	Name of every customer
customer_since	Date of customer's first purchase
gender	Gender of customer
birthyear	Birth year of every customer
generation	Generation of every customer based on birth year

Table 1.2: Attributes for 'Customer' data

3) Product

product_id	Unique ID for every product
product_group	Category of product
product	Product name
current_retail_price	Current product price
promo_yn	Product promotion availability
new_product_yn	New product availability

Table 1.3: Attributes for 'Product' data

1.3.1 Star Schema

We developed a star schema to understand the relationship between tables as well as to organize the data so that it is easy to understand and analyze.

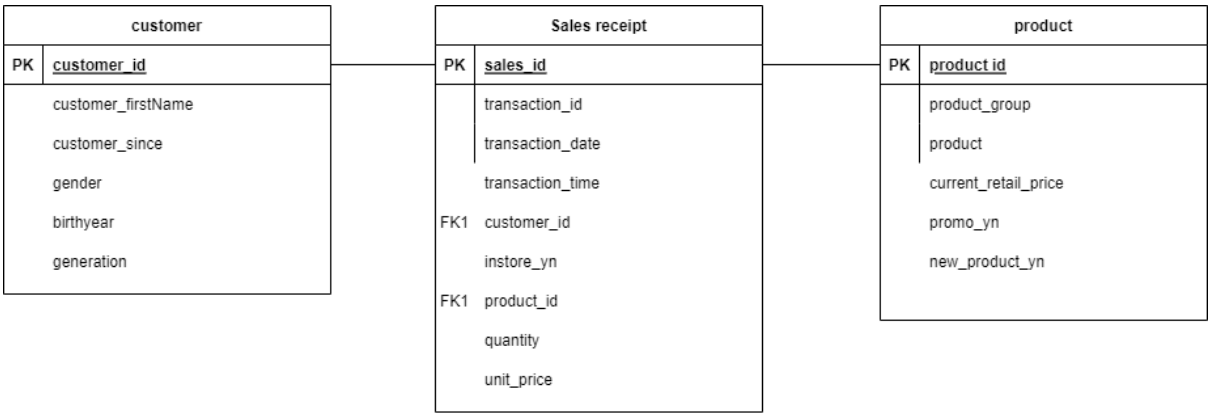


Diagram 1.1: Star Schema

1.3.2 Feature Tools

After designing Star Schema, we utilized *featuretools* library from Python to extract new features from current dataset. From a list of columns that have been extracted, we simply choose a few of them that are aligned with our business objectives.Below are the Python code developed to extract new features using Featuretools, and the list of features that have been extracted and chosen based on our business requirements.

```
import featuretools as ft
import pandas as pd

# Define data for Customers, Products, Orders, and Time
sales_df = pd.read_csv(r"C:\Users\User\OneDrive\Desktop\Tutorial UM\WIE3007 DATA
MINING AND WAREHOUSING\dataset\sales_receipts.csv")
customer_df = pd.read_csv(r"C:\Users\User\OneDrive\Desktop\Tutorial UM\WIE3007
DATA MINING AND WAREHOUSING\dataset\customer.csv")
product_df = pd.read_csv(r"C:\Users\User\OneDrive\Desktop\Tutorial UM\WIE3007 DATA
MINING AND WAREHOUSING\dataset\product.csv")

# Create an EntitySet
es = ft.EntitySet(id='coffee_data')

# Add dataframes as entities to the EntitySet
es.add_dataframe(dataframe_name='customer', dataframe=customer_df, index='
customer_id')
es.add_dataframe(dataframe_name='product', dataframe=product_df, index=' product_id')
es.add_dataframe(dataframe_name='sales', dataframe=sales_df, index=' sales_id')

# Define the relationships between entities
relationships = [
    ('customer', ' customer_id', 'sales', ' customer_id'),
    ('product', ' product_id', 'sales', ' product_id')
]

# Add the defined relationships to the EntitySet
for relationship in relationships:
    es = es.add_relationship(parent_dataframe_name=relationship[0],
                             parent_column_name=relationship[1],
                             child_dataframe_name=relationship[2],
                             child_column_name=relationship[3])


# Perform Deep Feature Synthesis with aggregation primitives
feature_matrix, feature_defs = ft.dfs(entityset=es, target_dataframe_name='sales',
                                      agg_primitives=['sum', 'mean', 'count', 'mode'],  # Add your
aggregation primitives
                                      trans_primitives=['month', 'weekday', 'day'],  # Add your
transformation primitives
                                      max_depth=2)

# Print the generated features
print(feature_matrix)
```

Python Source Code for Feature Extraction using Featuretools

Extracted column name	description
customer.COUNT(sales)	The quantity of products every consumer has bought. Importance: Analyzing customer behavior over generations depends on knowing how frequently a client makes purchases, which is something this tool helps with. Suitability: We can determine regular customers, infrequent purchasers, and the general level of interaction between various age groups and the items by knowing the number of sales made by each individual.
customer.MEAN(sales. IMP__quantity)	Average number of products a customer has bought. Significance: This feature offers insight into the common purchasing habits about quantity. Applicability: This helps customize marketing plans according to the typical amount that various generations purchase. For instance, companies may design promotions to encourage higher sales if a certain generation prefers to buy in bulk.
customer.MEAN(sales.unit_price)	Average amount of money spent with each customer Importance: This feature offers information on the average amount spent by each client. This allows companies to determine the average amount of money spent by each generation. Suitability: Marketing tactics may be adjusted to give targeted promotions by taking into consideration the average unit pricing. For example, companies can give superior product choices to an age group that has a tendency to purchase more expensive products.
customer.MODE(sales. IMP__instore_yn)	Highest frequency of the customer's approach to purchase Importance: The most commonly occurring value for the "instore_yn" variable, which indicates whether consumers prefer in-store or online transactions, is found using this feature tool.

	Suitability: Knowing the preferred mode of purchase is essential to adjusting marketing tactics. If a specific age group shops mostly online, we may enhance our online presence and concentrate on online promotions. This can assist in determining which things are more likely to be purchased physically and which more often online. In this manner, companies may concentrate on online advertising and enhance the digital presence if a specific product category is preferred to be purchased online. Companies may do live product marketing if an actual product is preferred.
customer.SUM(sales. IMP__quantity)	Total amount of goods purchased in quantity Importance: This tool computes the overall amount of products acquired by every customer, offering valuable information on the total amount of goods purchased. Suitability: Beneficial for evaluating the total demand from various age groups. Marketing tactics may need to provide special consideration to a generation if it makes a major contribution to the overall amount.
customer.SUM(sales.unit_price)	Total amount of money spent Important: This tool provides insights into the total income earned by various age groups by computing each customer's total expenditure. Suitability: Good for figuring out how much different generations can afford. You can customize promotions to the tastes of a certain age group if they represent a sizable portion of overall expenditure.
product.COUNT(sales)	Total sales of a product assigned by popularity. Importance: Gives the number of sales transactions for each product, which is helpful in determining the popularity of certain things (as the companies are figuring out how many people are purchasing them). Suitability: Beneficial for enhancing inventory management and product selection. Companies may change the selection and marketing techniques if a few products are more popular with particular age

	groups.
product.SUM(sales. IMP__quantity)	Total amount of highly desired products that have been sold. Importance: Determines the overall amount of a certain product sold, assisting in the comprehension of the product's demand. Suitability: Beneficial for maximizing stock levels and guaranteeing that popular products are sufficiently supplied during times of peak demand determined by examining daily buying trends .
product.SUM(sales.unit_price)	The total sales for every product Importance: To learn more about the overall amount of money brought in by a certain product. Companies may also learn about a product's overall income according to each generation. Suitability: Assists in comprehending the monetary value of various products. To increase sales, the company can concentrate on marketing items that appeal to particular age groups if that is where the majority of their purchases are coming from.

Table 1.4: List of new features from featuretools

1.3.3 Data Integration

We use Talend Data Integration to integrate three datasets i.e. sales receipt, customer, and product into a centralized dataset.

→ **Tool: Talend Data Integration**

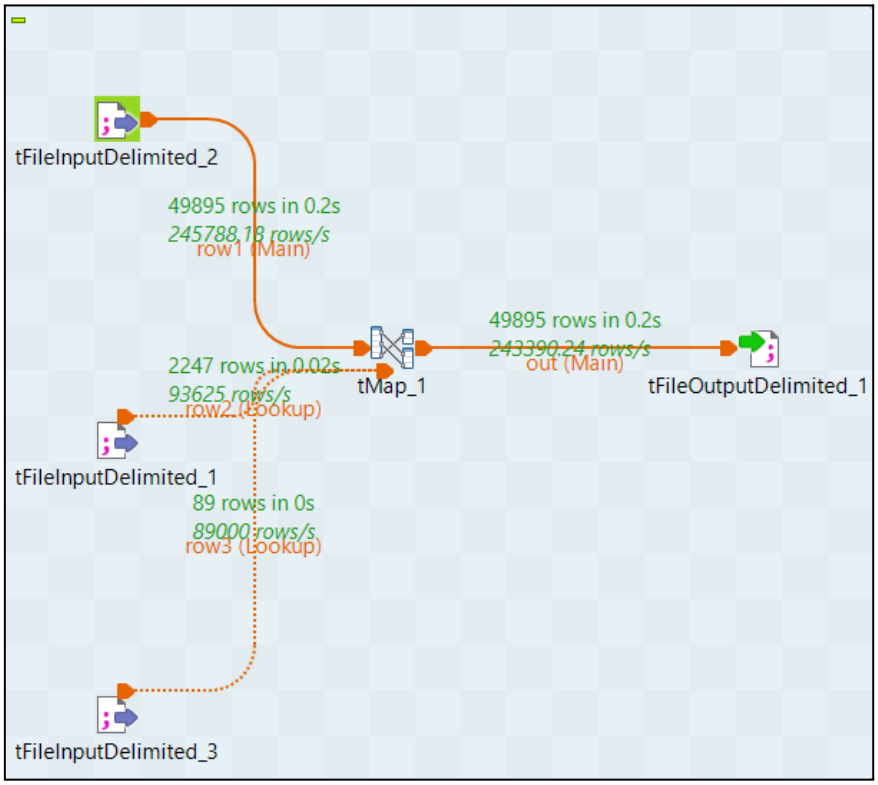


Diagram 1.2: Join three different datasets into one dataset.

On the left side of the tMap editor, we input three datasets and connect them with the tFileInputDelimited.

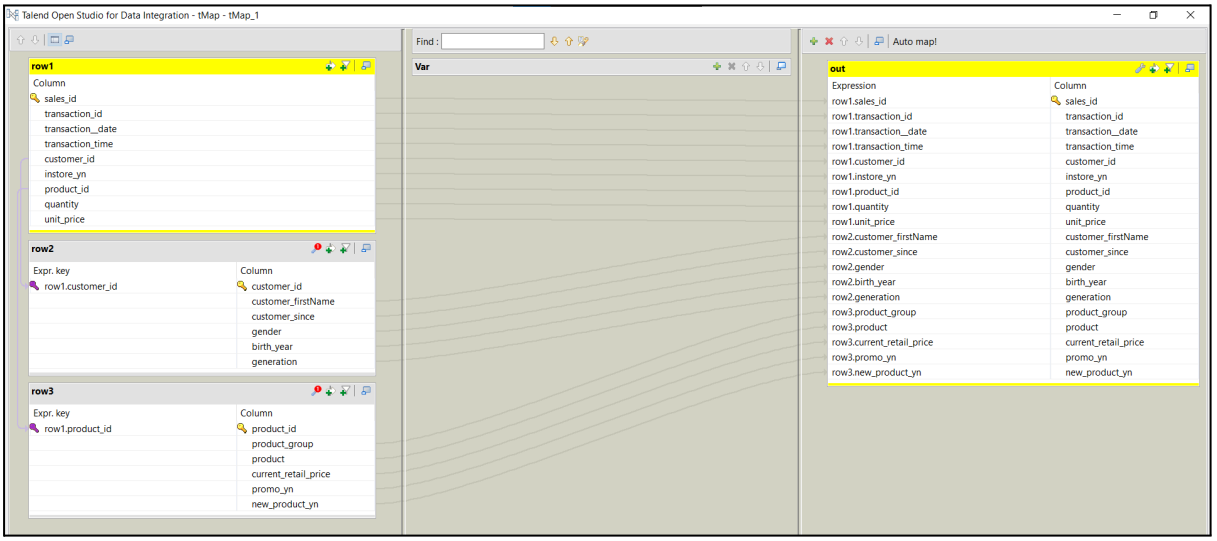


Diagram 1.3: Define output structure

We define the output structure to include all columns from table “sales_receipt” and select all the rows excluding “customer_id” and “product_id”.

2.0 SEMMA Methodology

2.1 Sample

Objective: To select a representative subset of data from the entire dataset.

This subset is used in the initial phases of the data mining process to make the computations more manageable and efficient. Moreover, sampling ensures that any patterns or relationships identified in the sample will be reflective of the entire dataset without losing the essential characteristics of the original dataset, making the process both time and resource efficient.

2.1.1 Sample Technique

We decided to use Random Sampling as the sample technique. Random sampling is a fundamental technique used in both statistics and data analysis, including data mining, to select a subset of observations from a larger population or data. Random sampling can give a general overview of the data without any biases. It's simple and can be a good starting point if there aren't any specific objectives. It is important to highlight that the key feature of random sampling is that each observation within the population has an equal probability of being selected for the sample.

Random sampling is employed to provide an unbiased and general overview of the data, thereby reducing the likelihood of selecting a non-representative sample that may introduce distorted or biased findings. It's simple and can be a good starting point as there aren't any specific objectives.

In this group project, the original dataset comprises a total of 49,894 rows. Then, we use SAS Enterprise Miner as the tool for random sampling. The sample configuration that has been applied is with a set percentage of 1%. Consequently, the result of the random sampling or known as subset of data consists of precisely 499 rows.

→ **Tool: SAS Enterprise Miner**

2.1.2 Target variable

From our dataset, we determine the target variables which are “Generation” and “Product”.

<u>Target Variable: Generation</u>

The "Generation" variable becomes an important factor in both our predictive models and analyses. Our goal is to develop models that accurately classify customers into various generational groups, such as Baby Boomers, Gen X, and others, by utilizing a large number of relevant characteristics and factors from our Coffee Shop dataset. By leveraging this predictive capability, the coffee corporation may adjust its marketing campaigns, product offerings, and operational decisions to the specific tastes and behaviors exhibited by individual generations. In order to ensure a seamless integration of its strategies with the fundamental business objective of generational customer behavior analysis, the organization must possess an extensive comprehension of generational dynamics.

It is important to acknowledge the "Generation" variable in order to customize marketing strategies and product offerings to accommodate the various interests and behaviors exhibited by each age group.

Target Variable: Product Group

The "Product Group" variable is important to our analysis, as it provides the fundamental structure for understanding the number and preferences of the varied beverages offered on the menu of the coffee shop. By incorporating this variable together with various components of the dataset, our aim is to develop models capable of accurately predicting the level of popularity of each product among different age groups. This predictive capability is consistent with the third business objective, which is to determine the product group's generational distribution. The results derived from this analysis will contribute to the enhancement of inventory management, the customization of the menu in response to different tastes of every age group, and the improvement of product selection.

The utilization of the "Product Group" variable facilitates the synchronization of promotional activities and the enhancement of resource allocation by considering the varying periods of demand for distinct beverages.

General	
Node ID	Smpl
Imported Data	...
Exported Data	...
Notes	...
Train	
Variables	...
Output Type	Data
Sample Method	Random
Random Seed	12345
Size	
Type	Percentage
Observations	.
Percentage	1.0
Alpha	0.01
PValue	0.01
Cluster Method	Random

Diagram 2.1.1: The settings of Random Sampling

Variable Summary

Role	Measurement Level	Frequency Count
ID	NOMINAL	4
INPUT	INTERVAL	4
INPUT	NOMINAL	8
TIMEID	INTERVAL	3

Sampling Summary

Type	Data Set	Number of Observations
DATA	EMWS3.FIMPORT_train	49894
SAMPLE	EMWS3.Smpl_DATA	499

Diagram 2.1.2: Result of Random Sampling

2.2 Explore

Explore phase in the SAS SEMMA methodology is a critical step that involves a detailed examination of the sampled data. This phase is essential for gaining a deeper understanding of the dataset, revealing underlying patterns, relationships, anomalies, and trends.

Exploration helps identify underlying structures and characteristics by offering a deeper understanding of the data. This deeper understanding contributes to the development of effective models and assists in addressing possible problems in the dataset.

By using SAS Enterprise Miner, we explore the sample dataset using three types of techniques which are descriptive statistics, data visualization and correlation analysis which will be explained further in the next section.

2.2.1 Descriptive Statistics

The main task of descriptive statistics is to provide an overview and description of a dataset's key characteristics. For every variable in the sampled data, a variety of summary measures must be computed and analyzed. The main objective is to provide a concise yet clear overview of the dataset's features.

In this group project, we identify two main types of descriptive statistics which are central tendency and variability. Central tendency concerns the averages of the values, indicating the central or typical value around which the data tends to cluster while variability concerns how spread out the values are, providing information about the degree of variation or scatter in the dataset. This aspect helps understand the range of values and how they deviate from the central tendency.

The key measures under central tendency are mean and median while the key measures for variability are standard deviation, minimum and maximum. We also identify the number of missing values that exist in the sample dataset which is 253 for each of customer_firstName, gender, and generation, while 3 missing values exist for instore_yn.

Key Measure	Definition	Purpose
Mean	The sum of all values divided by the number of values.	Indicates the central tendency or the "typical" value in the dataset.
Standard Deviation	A measure of the amount of variation or dispersion in a set of values.	Indicates the spread or variability around the mean.
Minimum	The smallest value in the sample dataset.	Highlights the lower boundary or limit of the sample dataset.
Median	The middle value in a dataset	Provides a measure of central

	when arranged in ascending or descending order.	tendency, robust to extreme values or outliers.
Maximum	The largest value in the sample dataset.	Highlights the upper boundary or extreme values in the sample dataset.

Table 2.2.1: Definition and Purpose of each Key Measure

The summarization of descriptive statistics is shown in the diagram below.

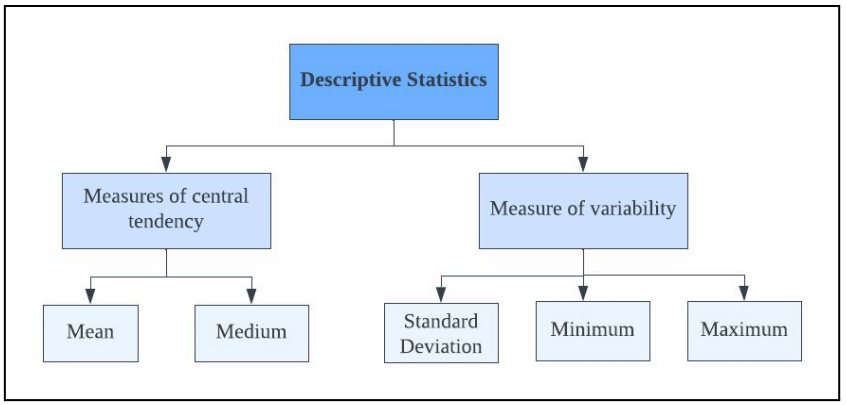


Diagram 2.2.1: Descriptive Statistics

Variable Summary

Role	Measurement Level	Frequency Count
ID	NOMINAL	4
INPUT	INTERVAL	5
INPUT	NOMINAL	9

Variable Levels Summary
(maximum 500 observations printed)

Variable	Role	Frequency Count
_customer_id	ID	233
_product_id	ID	70
_transaction_id	ID	431
sales_id	ID	499

Class Variable Summary Statistics
(maximum 500 observations printed)

Data Role=TRAIN

Data Role	Variable Name	Role	Number of Levels	Missing	Mode	Mode Percentage	Mode2	Mode2 Percentage
TRAIN	VAR21	INPUT	1	499		100.0		0.00
TRAIN	_customer_firstName	INPUT	222	0	Alexa Hancock	4.81	Giacomo Luna	3.01
TRAIN	_gender	INPUT	3	0	M	42.89	F	42.08
TRAIN	_generation	INPUT	5	0	Baby Boomers	36.67	Gen X	20.44
TRAIN	_instore_yn	INPUT	3	3	N	49.90	Y	49.50
TRAIN	_new_product_yn	INPUT	1	0	N	100.0		0.00
TRAIN	_product	INPUT	70	0	Morning Sunrise Chai Rg	3.21	Earl Grey Rg	3.01
TRAIN	_product_group	INPUT	5	0	Beverages	76.35	Food	13.83
TRAIN	_promo_yn	INPUT	1	0	N	100.0		0.00

Interval Variable Summary Statistics
(maximum 500 observations printed)

Data Role=TRAIN

Variable	Role	Mean	Standard Deviation	Non Missing	Missing	Minimum	Median	Maximum	Skewness	Kurtosis
__dataobs_	INPUT	25735.74	14818.52	499	0	151	24850	49883	-0.02326	-1.28605
_birth_year	INPUT	1974.922	15.54072	499	0	1950	1972	2001	0.132874	-1.39266
_current_retail_price	INPUT	3.739479	3.060348	499	0	1	3	28	5.127938	30.13735
_quantity	INPUT	1.447423	0.556534	485	14	1	1	3	0.750323	-0.49067
_unit_price	INPUT	3.567154	3.068847	499	0	0.8	3	28	5.158455	30.32614

Diagram 2.2.2: Result of StatExplore in SAS Enterprise Miner

2.2.2 Data Visualization

Data exploration and analysis are greatly assisted by data visualization, which enables the graphical representation of data and information. It is possible to use it to represent, among other things, one, two, or three variables, each of which serves a unique function and offers distinct perspectives.

In this group project, we use two-variable visualization (Bivariate) which is represented by a box plot and three-variable visualization (Trivariate) which is represented by a scatter plot. The further explanation of the visualization is in the next section.

2.2.2.1 Two-Variable Visualization (Bivariate)

The objective of bivariate visualization is to explore the relationship, correlation, or interaction between two variables. Bivariate involves the simultaneous representation of two variables, allowing for the exploration of relationships and patterns between them. A scatter plot is a common and effective bivariate visualization tool.

→ **Tool: SAS Enterprise Miner**

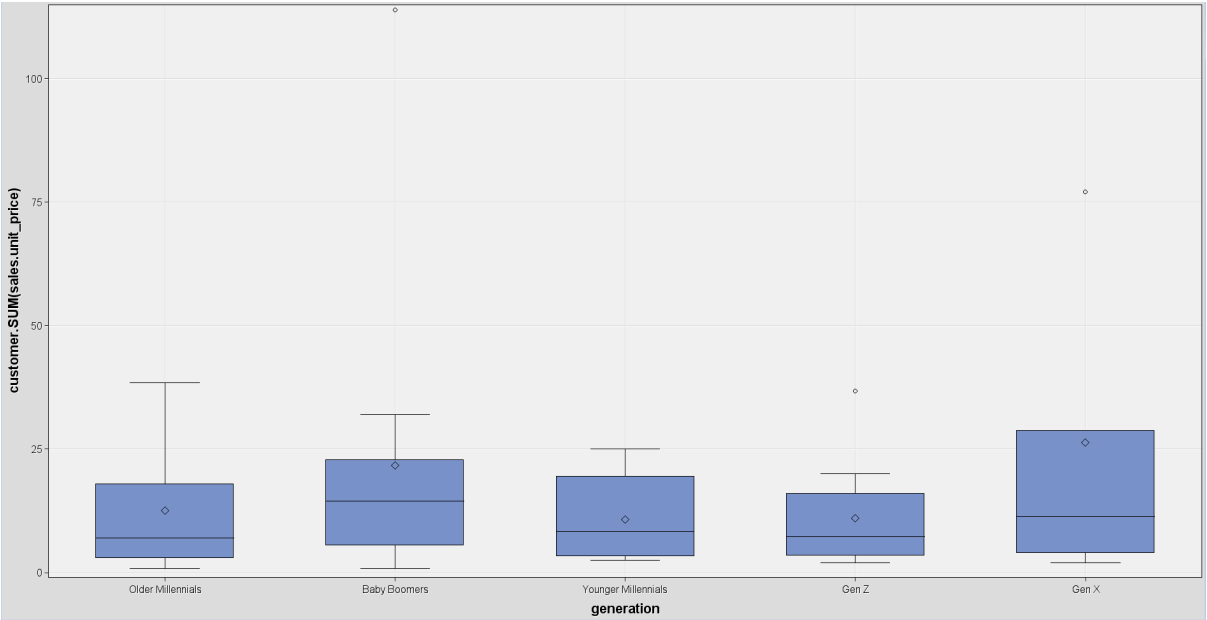


Diagram 2.2.3: A box plot showing the customer.SUM(sales.unit_price) and generation.

For the diagram above, we can see that Baby Boomers have the highest median value, while the older millennials have the lowest median, that shows Baby Boomers spending much more on that contribution to the sales. While Gen X has the highest concentration on expenditure towards sales.

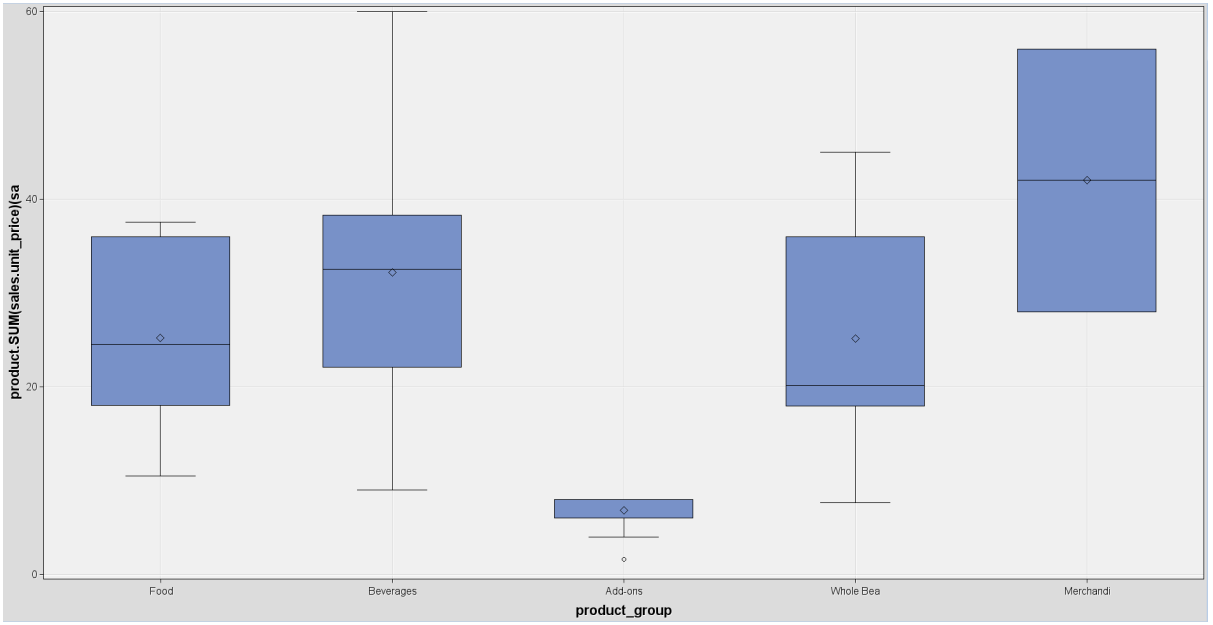


Diagram 2.2.4: A box plot showing the distribution of product.SUM(sales.unit_price) and product_group.

In product.SUM(sales.unit_price) and product_group, beverages get the highest sales compared to other products, while Merchandise has the highest number of total sales in product.

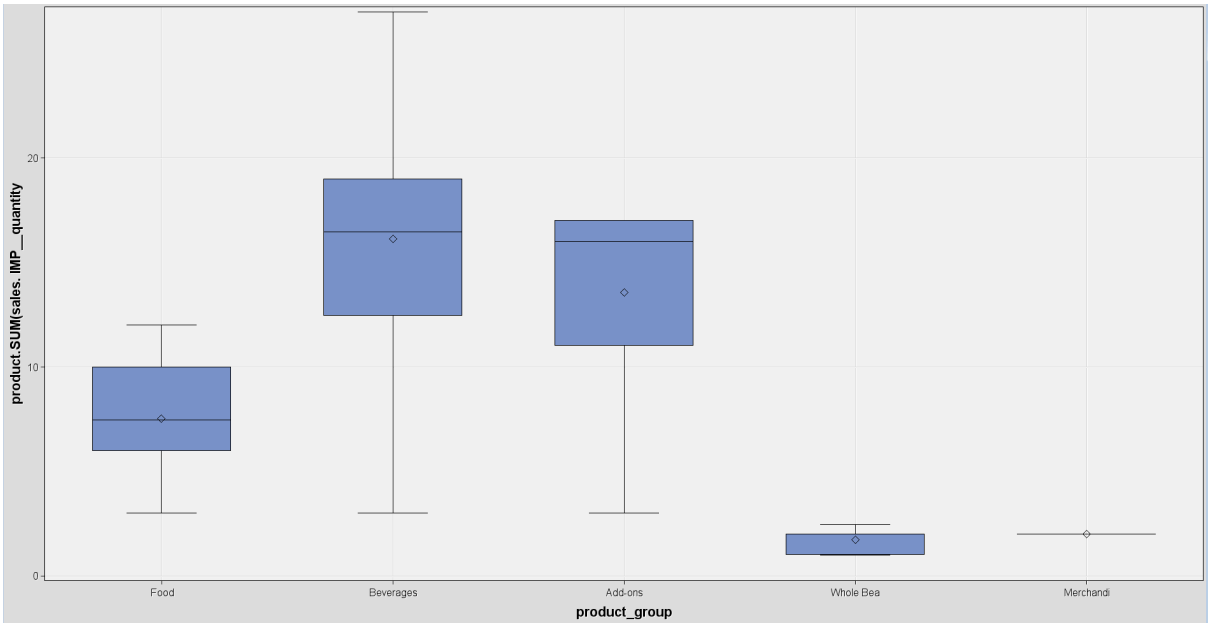


Diagram 2.2.5: A box plot showing the distribution of product.SUM(sales. IMP__quantity) and product_group.

For the aspect product.SUM(sales. IMP__quantity) and product_group, Beverages has the highest maximum quantity on the product, while Whole Bean has the lowest minimum quantity on the product group.

2.2.2.2 Three-Variable Visualization (Trivariate)

The primary goal of trivariate visualization is to gain insights into the interactions and relationships among three variables simultaneously. This method enables a more thorough comprehension of the relationships and mutual influences between these variables.

For this group project, we create a scatter plot chart, in which the color indicates the generation group and the x- and y-axes stand for numerical qualities.

→ **Tool: Jupyter (Python)**

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Read the CSV file into a DataFrame
coffee_data = pd.read_csv(r"C:\Users\User\OneDrive\Desktop\Tutorial UM\WIE3007 DATA MINING AND WAREHOUSING\dataset\coffee_dataset_featuretools.csv")

columns_to_drop = ['sales_id', 'transaction_id', 'customer_id', 'product_id'] # Replace with the actual column names you want to drop
coffee_data = coffee_data.drop(columns=columns_to_drop)

# Select specific columns and drop rows with missing values
coffee_data = coffee_data.dropna()

# Set Seaborn settings
sns.set(style="ticks", font_scale=0.7, rc={"xtick.labelsize": 7, "ytick.labelsize": 7})

pair_plot = sns.pairplot(coffee_data, hue='generation', diag_kind="kde", markers=["o", "s", "D", "P", "X"], height=2)

# Save the plot as an image file
pair_plot.savefig(r"C:\Users\User\OneDrive\Desktop\Tutorial UM\WIE3007 DATA MINING AND WAREHOUSING\img_coffee_by_generation.png")

# Show the plot
plt.show()
```

Python Source Code for generating scatter plot for each interval variables

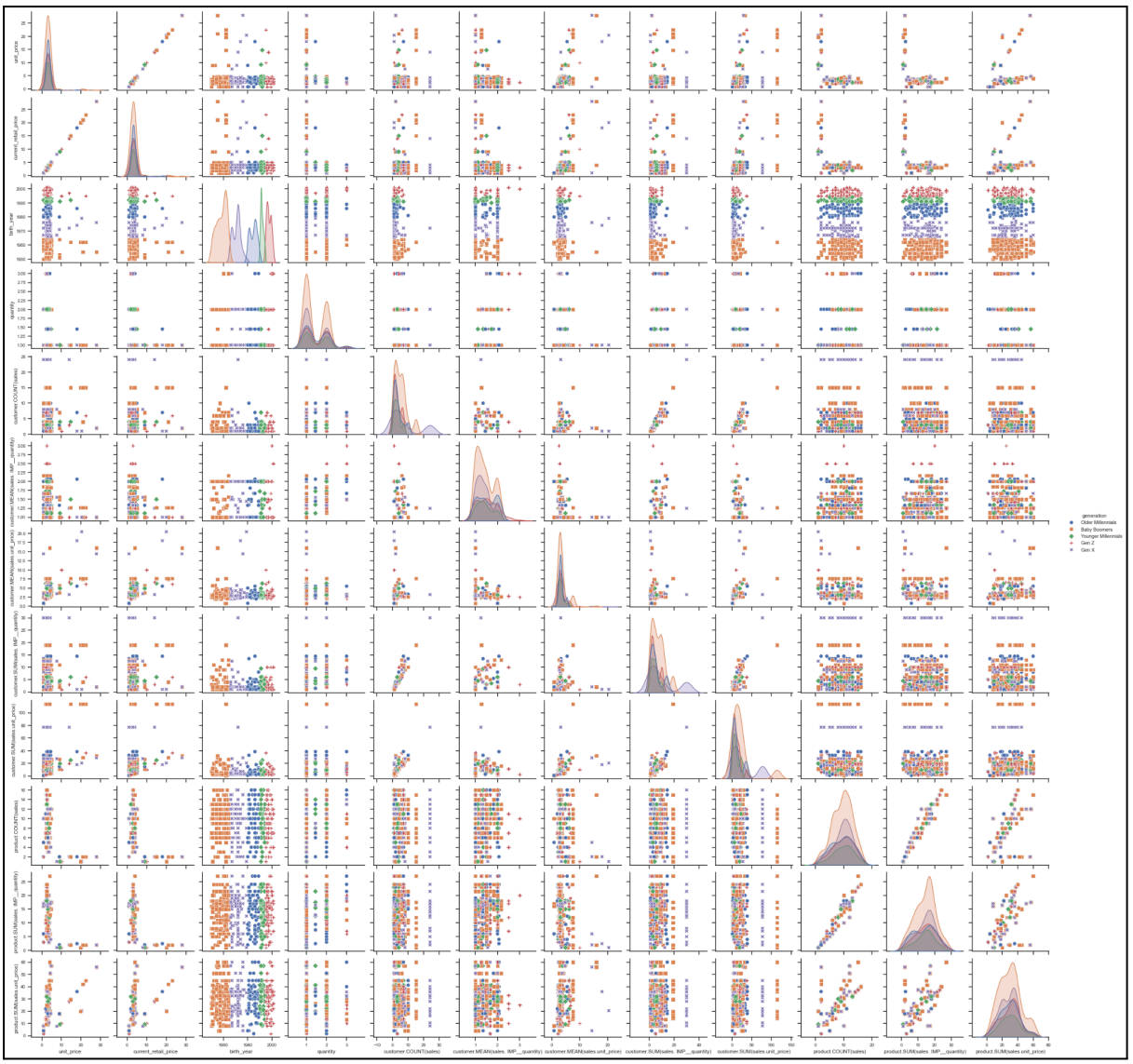


Diagram 2.2.6: Scatter plot showing the relationship between attributes in the sample dataset.

2.2.3 Correlation Analysis

Studying and measuring the strength of the association between two or more variables is the goal of correlation analysis. By examining relationships and dependencies, this study aims to determine whether and how modifications in one variable are associated with modifications in another. We intend to find out the distribution of each generation group on each numerical variable for the purpose of this task. Furthermore, we are interested in examining the correlation that exists among each numerical variable.

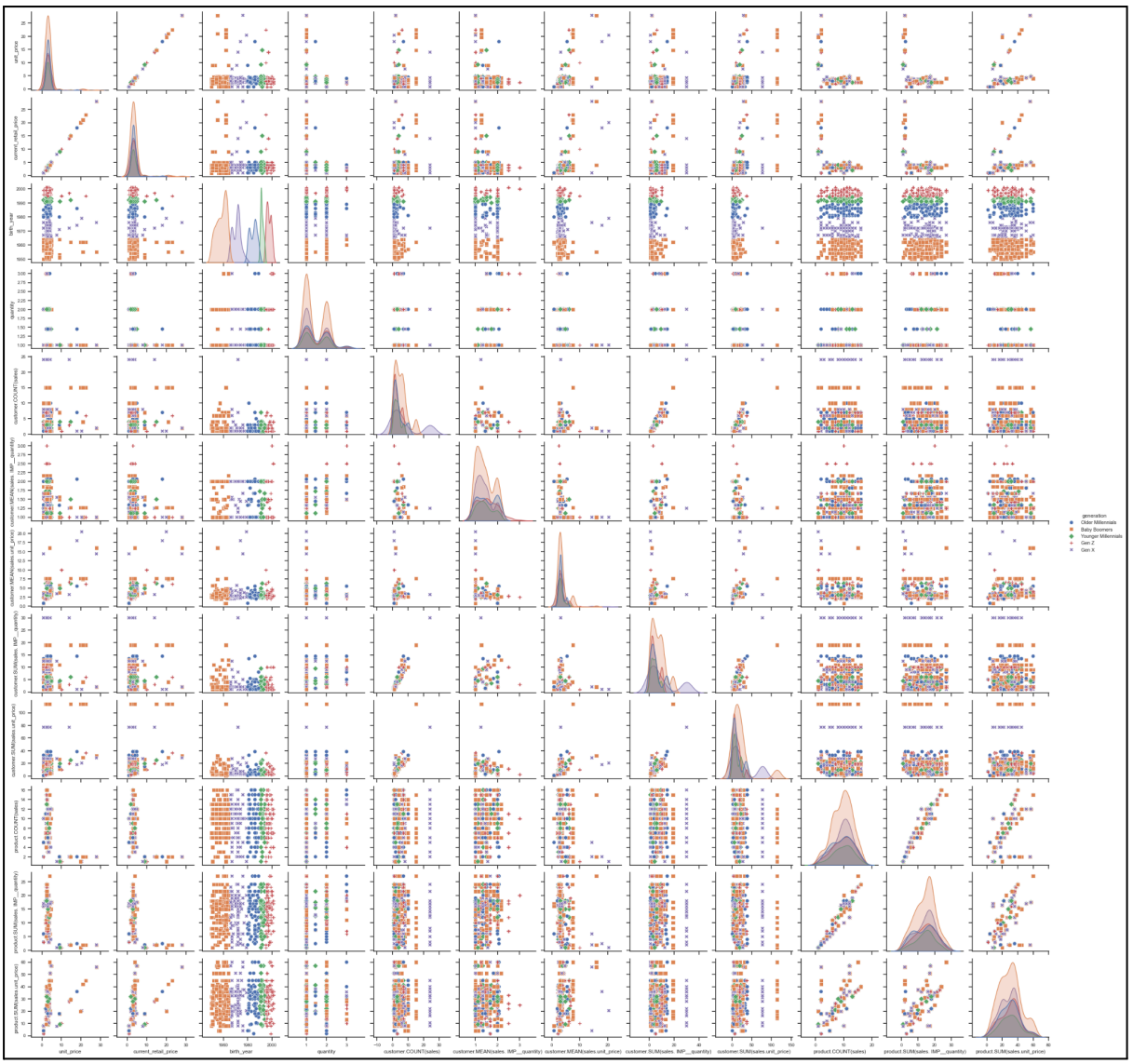


Diagram 2.2.7: Scatter plot showing the relationship between attributes in the sample dataset.

From the correlation analysis above, we identified that the group of generation (color) plays a significant role in variables customer.COUNT(sales), customer.SUM(sales.quantity), and customer.SUM(sales.unit_price). For example, in both customer.COUNT(sales) and customer.SUM(sales.quantity), Gen X appears to be the generation group with highest value, while for customer.SUM(sales.unit_price), it would be the generation group Baby Boomers.

Apart from that, when determining the correlation between variables, we identified two highly correlated variables which are between unit_price and current_retail_price, and between product.COUNT(sales) and product.SUM(sales.IMP_quantity). Therefore, one of the variables from each need to be dropped before fitting the data into the model.

2.2.4 Association Analysis

Association analysis, akin to correlation analysis in statistical techniques, emerges as a crucial instrument in the Explore stage of data warehouse development. This method delves into vast datasets to unearth concealed relationships, patterns, and insights.

In this group project, we do Association Rules and Sequence Rules using the node of 'Association' in SAS Enterprise Miner as below.

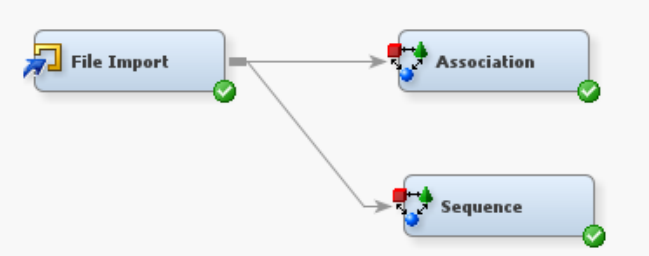


Diagram 2.2.8: Association Analysis Nodes in SAS Enterprise Miner

2.2.4.1 Association Rules

Association rules are patterns or relationships discovered in datasets using association analysis or association rule mining techniques. The basic idea is to find interesting relationships or patterns in the form of rules in order to make predictions, comprehend consumer behavior, or obtain insights into the underlying data.

In the association node, the ID variable is assigned to customer_id, and the Target variable is assigned to product. Diagram below is the result of the association analysis.

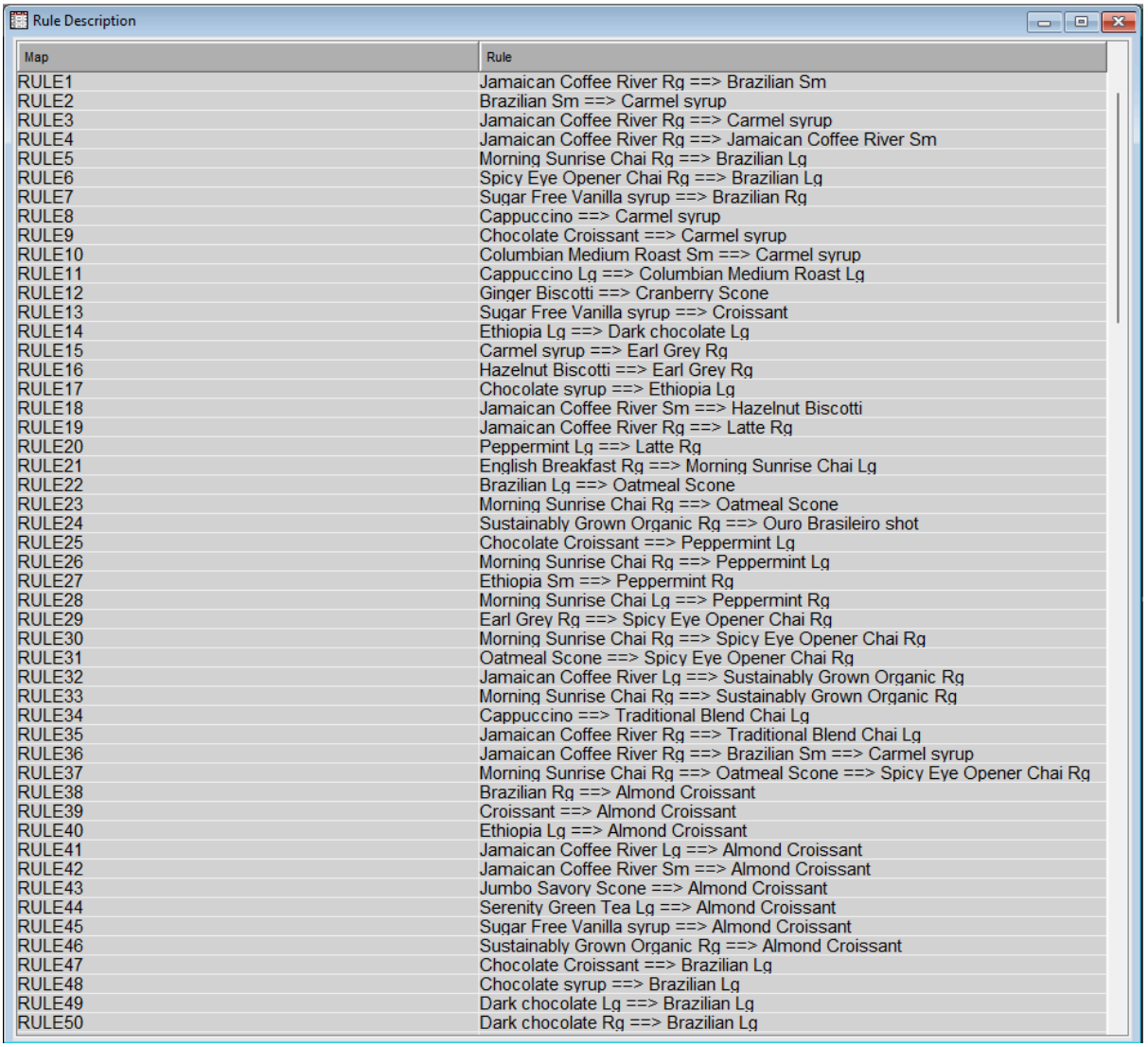
Map	Rule
RULE1	Sustainably Grown Organic Rg & Sugar Free Vanilla syrup ==> Brazilian Rg & Almond Croissant
RULE2	Sustainably Grown Organic Rg & Lemon Grass Rg ==> Brazilian Rg & Almond Croissant
RULE3	Sustainably Grown Organic Rg & Jumbo Savory Scone ==> Brazilian Rg & Almond Croissant
RULE4	Sustainably Grown Organic Rg & Ethiopia Lg ==> Brazilian Rg & Almond Croissant
RULE5	Sustainably Grown Organic Rg & Dark chocolate Lg ==> Brazilian Rg & Almond Croissant
RULE6	Sustainably Grown Organic Rg & Croissant ==> Brazilian Rg & Almond Croissant
RULE7	Sugar Free Vanilla syrup & Lemon Grass Rg ==> Brazilian Rg & Almond Croissant
RULE8	Sugar Free Vanilla syrup & Jumbo Savory Scone ==> Brazilian Rg & Almond Croissant
RULE9	Sugar Free Vanilla syrup & Jamaican Coffee River Lg ==> Brazilian Rg & Almond Croissant
RULE10	Sugar Free Vanilla syrup & Ethiopia Lg ==> Brazilian Rg & Almond Croissant
RULE11	Sugar Free Vanilla syrup & Dark chocolate Lg ==> Brazilian Rg & Almond Croissant
RULE12	Lemon Grass Rg & Jumbo Savory Scone ==> Brazilian Rg & Almond Croissant
RULE13	Lemon Grass Rg & Jamaican Coffee River Lg ==> Brazilian Rg & Almond Croissant
RULE14	Lemon Grass Rg & Croissant ==> Brazilian Rg & Almond Croissant
RULE15	Jumbo Savory Scone & Jamaican Coffee River Lg ==> Brazilian Rg & Almond Croissant
RULE16	Jumbo Savory Scone & Ethiopia Lg ==> Brazilian Rg & Almond Croissant
RULE17	Jumbo Savory Scone & Croissant ==> Brazilian Rg & Almond Croissant
RULE18	Jamaican Coffee River Lg & Ethiopia Lg ==> Brazilian Rg & Almond Croissant
RULE19	Jamaican Coffee River Lg & Dark chocolate Lg ==> Brazilian Rg & Almond Croissant
RULE20	Jamaican Coffee River Lg & Croissant ==> Brazilian Rg & Almond Croissant
RULE21	Ethiopia Lg & Croissant ==> Brazilian Rg & Almond Croissant
RULE22	Dark chocolate Lg & Croissant ==> Brazilian Rg & Almond Croissant
RULE23	Traditional Blend Chai Lg & Peppermint Rg ==> Cappuccino & Brazilian Sm
RULE24	Traditional Blend Chai Lg & Ethiopia Sm ==> Cappuccino & Brazilian Sm
RULE25	Traditional Blend Chai Lg & Carmel syrup ==> Cappuccino & Brazilian Sm
RULE26	Peppermint Rg & Jamaican Coffee River Rg ==> Cappuccino & Brazilian Sm
RULE27	Peppermint Rg & Carmel syrup ==> Cappuccino & Brazilian Sm
RULE28	Jamaican Coffee River Rg & Ethiopia Sm ==> Cappuccino & Brazilian Sm
RULE29	Ethiopia Sm & Carmel syrup ==> Cappuccino & Brazilian Sm
RULE30	Jumbo Savory Scone & Ginger Scone ==> Cappuccino Lg & Almond Croissant
RULE31	Jumbo Savory Scone & Columbian Medium Roast Sm ==> Cappuccino Lg & Almond Croissant
RULE32	Jumbo Savory Scone & Chocolate Croissant ==> Cappuccino Lg & Almond Croissant
RULE33	Jumbo Savory Scone & Carmel syrup ==> Cappuccino Lg & Almond Croissant
RULE34	Ginger Scone & Columbian Medium Roast Sm ==> Cappuccino Lg & Almond Croissant
RULE35	Ginger Scone & Chocolate Croissant ==> Cappuccino Lg & Almond Croissant
RULE36	Ginger Scone & Carmel syrup ==> Cappuccino Lg & Almond Croissant
RULE37	Sustainably Grown Organic Rg & Earl Grey Rg ==> Cappuccino Lg & Brazilian Sm
RULE38	Sustainably Grown Organic Rg & Carmel syrup ==> Cappuccino Lg & Brazilian Sm
RULE39	Jumbo Savory Scone & Ginger Scone ==> Carmel syrup & Almond Croissant
RULE40	Jumbo Savory Scone & Columbian Medium Roast Sm ==> Carmel syrup & Almond Croissant
RULE41	Jumbo Savory Scone & Chocolate Croissant ==> Carmel syrup & Almond Croissant
RULE42	Jumbo Savory Scone & Cappuccino Lg ==> Carmel syrup & Almond Croissant
RULE43	Ginger Scone & Columbian Medium Roast Sm ==> Carmel syrup & Almond Croissant
RULE44	Ginger Scone & Chocolate Croissant ==> Carmel syrup & Almond Croissant
RULE45	Ginger Scone & Cappuccino Lg ==> Carmel syrup & Almond Croissant
RULE46	Columbian Medium Roast Sm & Cappuccino Lg ==> Carmel syrup & Almond Croissant
RULE47	Chocolate Croissant & Cappuccino Lg ==> Carmel syrup & Almond Croissant
RULE48	Sustainably Grown Organic Lg & Sugar Free Vanilla syrup ==> Chocolate Chip Biscotti & Brazilian Rg
RULE49	Sustainably Grown Organic Lg & Spicy Eye Opener Chai Rg ==> Chocolate Chip Biscotti & Brazilian Rg
RULE50	Sustainably Grown Organic Lg & Dark chocolate Rg ==> Chocolate Chip Biscotti & Brazilian Rg

Diagram 2.2.9: Association Rules results

2.2.4.2 Sequence Rules

Sequence rules are patterns or regulations that define the sequential relationships and temporal ordering of occurrences within a given dataset. These processes are also designated as sequential pattern mining. Sequence rules identify patterns by considering the order of occurrences, as opposed to association rules that emphasize co-occurrence.

In the sequence node, the ID variable is assigned to customer_id, the Target variable is assigned to product, and the Sequence variable is assigned to transaction_time. Diagram below is the result of the sequence analysis.



Map	Rule
RULE1	Jamaican Coffee River Rg ==> Brazilian Sm
RULE2	Brazilian Sm ==> Carmel syrup
RULE3	Jamaican Coffee River Rg ==> Carmel syrup
RULE4	Jamaican Coffee River Rg ==> Jamaican Coffee River Sm
RULE5	Morning Sunrise Chai Rg ==> Brazilian Lg
RULE6	Spicy Eye Opener Chai Rg ==> Brazilian Lg
RULE7	Sugar Free Vanilla syrup ==> Brazilian Rg
RULE8	Cappuccino ==> Carmel syrup
RULE9	Chocolate Croissant ==> Carmel syrup
RULE10	Columbian Medium Roast Sm ==> Carmel syrup
RULE11	Cappuccino Lg ==> Columbian Medium Roast Lg
RULE12	Ginger Biscotti ==> Cranberry Scone
RULE13	Sugar Free Vanilla syrup ==> Croissant
RULE14	Ethiopia Lg ==> Dark chocolate Lg
RULE15	Carmel syrup ==> Earl Grey Rg
RULE16	Hazelnut Biscotti ==> Earl Grey Rg
RULE17	Chocolate syrup ==> Ethiopia Lg
RULE18	Jamaican Coffee River Sm ==> Hazelnut Biscotti
RULE19	Jamaican Coffee River Rg ==> Latte Rg
RULE20	Peppermint Lg ==> Latte Rg
RULE21	English Breakfast Rg ==> Morning Sunrise Chai Lg
RULE22	Brazilian Lg ==> Oatmeal Scone
RULE23	Morning Sunrise Chai Rg ==> Oatmeal Scone
RULE24	Sustainably Grown Organic Rg ==> Ouro Brasileiro shot
RULE25	Chocolate Croissant ==> Peppermint Lg
RULE26	Morning Sunrise Chai Rg ==> Peppermint Lg
RULE27	Ethiopia Sm ==> Peppermint Rg
RULE28	Morning Sunrise Chai Lg ==> Peppermint Rg
RULE29	Earl Grey Rg ==> Spicy Eye Opener Chai Rg
RULE30	Morning Sunrise Chai Rg ==> Spicy Eye Opener Chai Rg
RULE31	Oatmeal Scone ==> Spicy Eye Opener Chai Rg
RULE32	Jamaican Coffee River Lg ==> Sustainably Grown Organic Rg
RULE33	Morning Sunrise Chai Rg ==> Sustainably Grown Organic Rg
RULE34	Cappuccino ==> Traditional Blend Chai Lg
RULE35	Jamaican Coffee River Rg ==> Traditional Blend Chai Lg
RULE36	Jamaican Coffee River Rg ==> Brazilian Sm ==> Carmel syrup
RULE37	Morning Sunrise Chai Rg ==> Oatmeal Scone ==> Spicy Eye Opener Chai Rg
RULE38	Brazilian Rg ==> Almond Croissant
RULE39	Croissant ==> Almond Croissant
RULE40	Ethiopia Lg ==> Almond Croissant
RULE41	Jamaican Coffee River Lg ==> Almond Croissant
RULE42	Jamaican Coffee River Sm ==> Almond Croissant
RULE43	Jumbo Savory Scone ==> Almond Croissant
RULE44	Serenity Green Tea Lg ==> Almond Croissant
RULE45	Sugar Free Vanilla syrup ==> Almond Croissant
RULE46	Sustainably Grown Organic Rg ==> Almond Croissant
RULE47	Chocolate Croissant ==> Brazilian Lg
RULE48	Chocolate syrup ==> Brazilian Lg
RULE49	Dark chocolate Lg ==> Brazilian Lg
RULE50	Dark chocolate Rg ==> Brazilian Lg

Diagram 2.2.10: Sequence Analysis results

2.2.5 Time-Series Analysis

Time-series analysis is a statistical method that identifies patterns, trends, and behaviors through the investigation and assessment of data points collected over a period of time. We shifted our focus to time series analysis in pursuit of the objectives of our collaborative attempt concerning data extraction and warehousing.

We looked to identify trends in consumer behavior and product appeal by using generations and product categories as Cross IDs. Initial results point to complex relationships between product preferences and age cohorts, which are insightful.

Diagram 2.2.11: Nodes involved in Time Series Analysis

Cross ID: Product Group

TSID Map Table

Time Series ID	Original Variable Name	Role	Variable Label	group
1	TS_1	TARGET	SALES 1	... Add-ons
2	TS_2	TARGET	SALES 2	... Beverages
3	TS_3	TARGET	SALES 3	... Food
4	TS_4	TARGET	SALES 4	... WholeBea

Table 2.2.2: TSID Map Table for Product Group

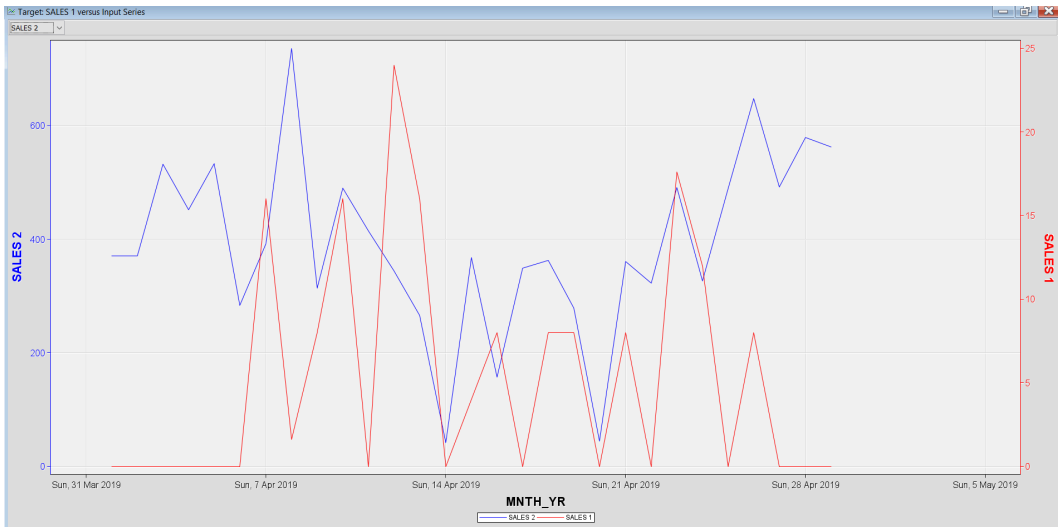


Diagram 2.2.12: Time Series Analysis of Sales1 vs Sales2

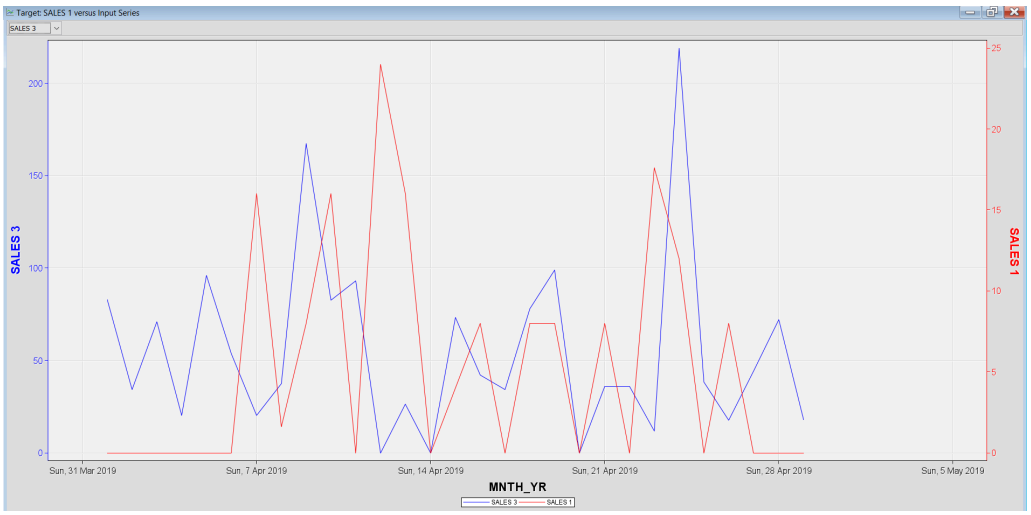


Diagram 2.2.13: Time Series Analysis of Sales1 vs Sales3

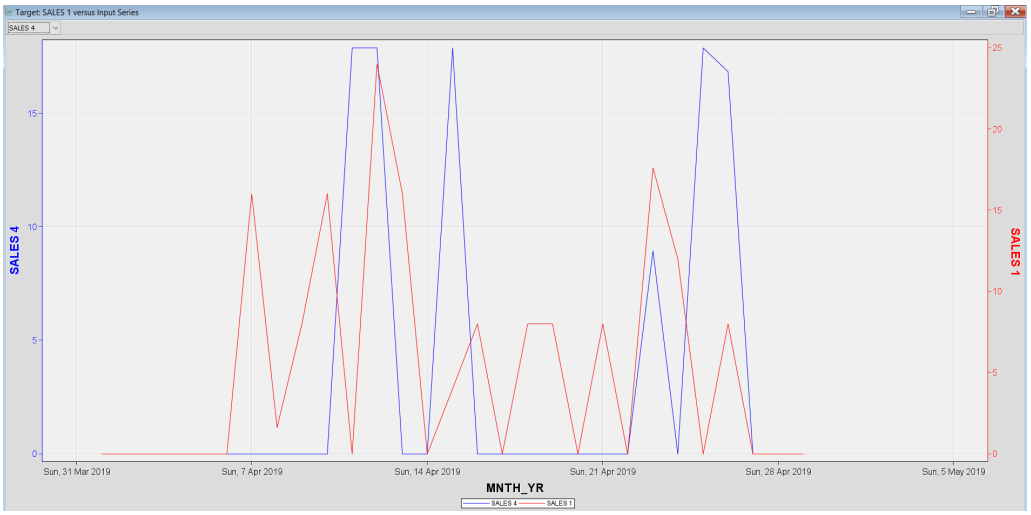


Diagram 2.2.14: Time Series Analysis of Sales1 vs Sales4

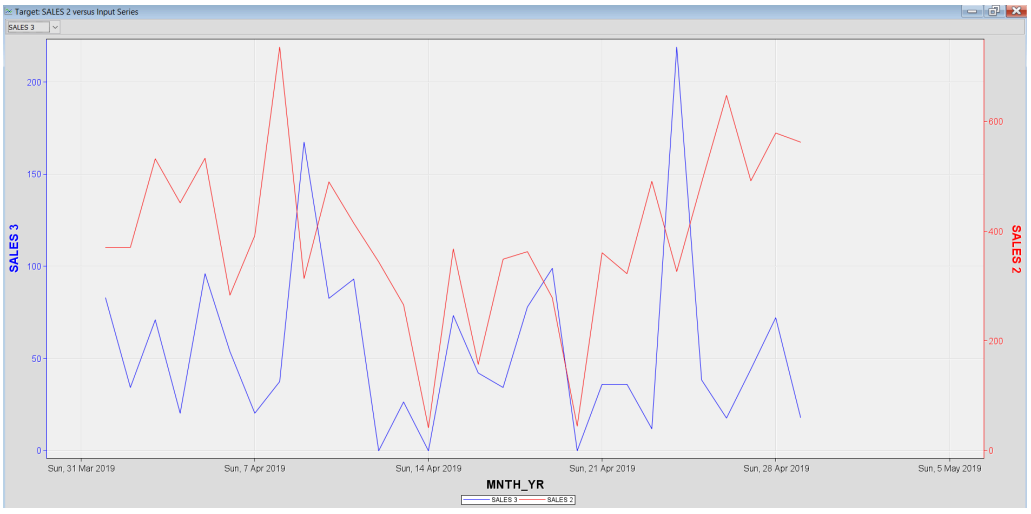


Diagram 2.2.15: Time Series Analysis of Sales2 vs Sales3

Diagram 2.2.16: Time Series Analysis of Sales2 vs Sales4

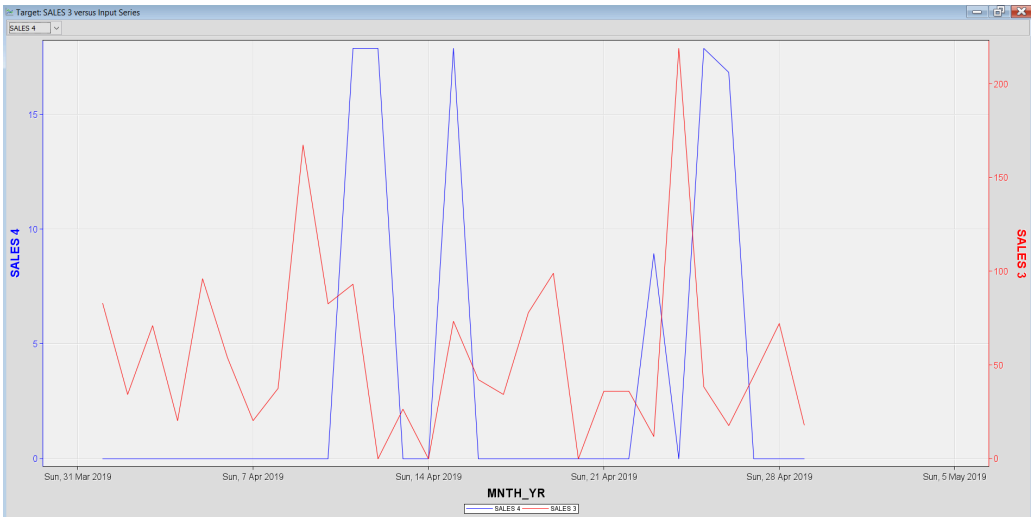


Diagram 2.2.17: Time Series Analysis of Sales3 vs Sales4

Based on this analysis, we can make the daily time series comparison between each product groups. The insight provides understanding on each product group's popularity within the given time interval.

For example, from the latest diagram, we can see at *Sun, 14 Apr 2019*, 'WholeBeans' product group (Sales 4) is more popular than the 'food' product group (Sales 3).

Cross ID: Generations

TSID Map Table

Time Series ID	Original Variable Name	Role	Variable Label	group
1	TS_1	TARGET	SALES 1	... BabyBoomers
2	TS_2	TARGET	SALES 2	... GenX
3	TS_3	TARGET	SALES 3	... GenZ
4	TS_4	TARGET	SALES 4	... OlderMillennials
5	TS_5	TARGET	SALES 5	... YoungerMillennials

Table 2.2.3: TSID Map Table for Generation Group

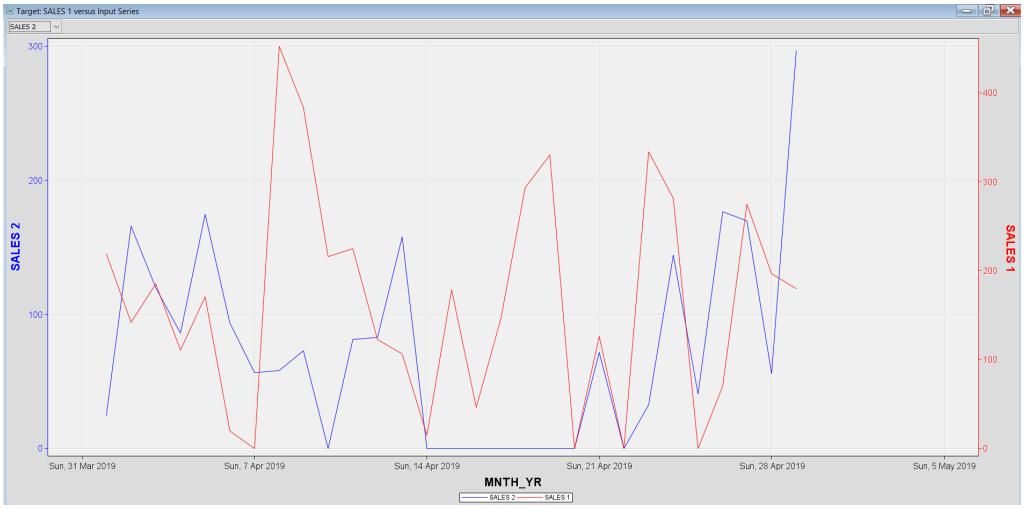


Diagram 2.2.18: Time Series Analysis of Sales1 vs Sales2

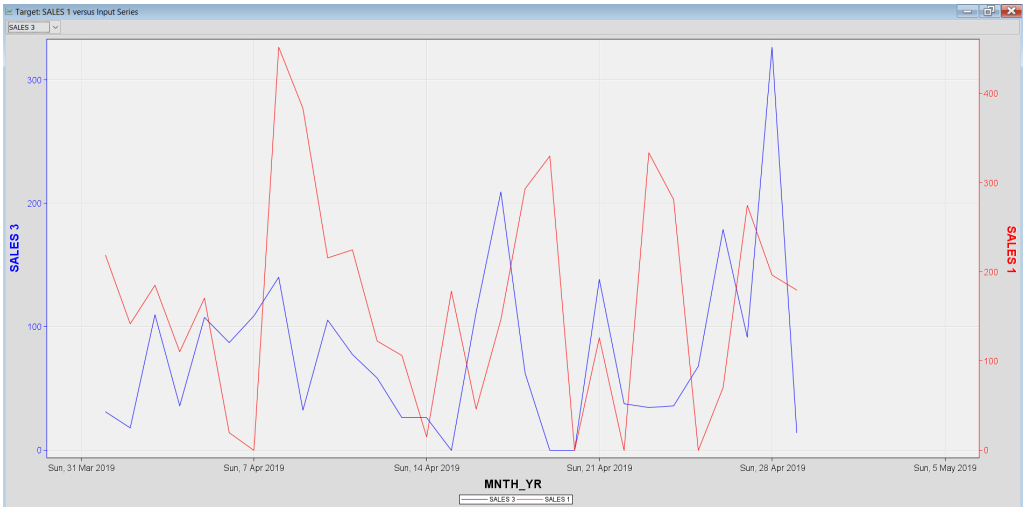


Diagram 2.2.19: Time Series Analysis of Sales1 vs Sales3

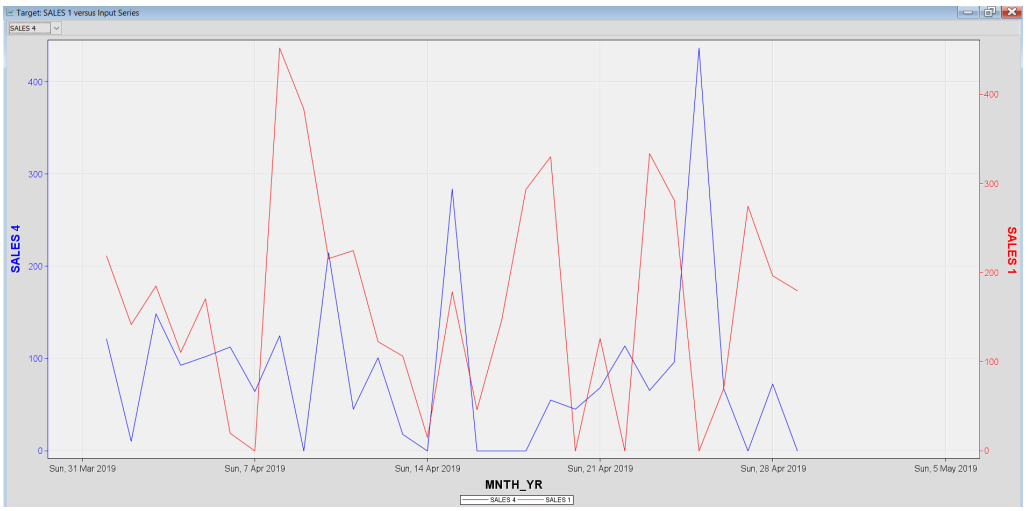


Diagram 2.2.20: Time Series Analysis of Sales1 vs Sales4

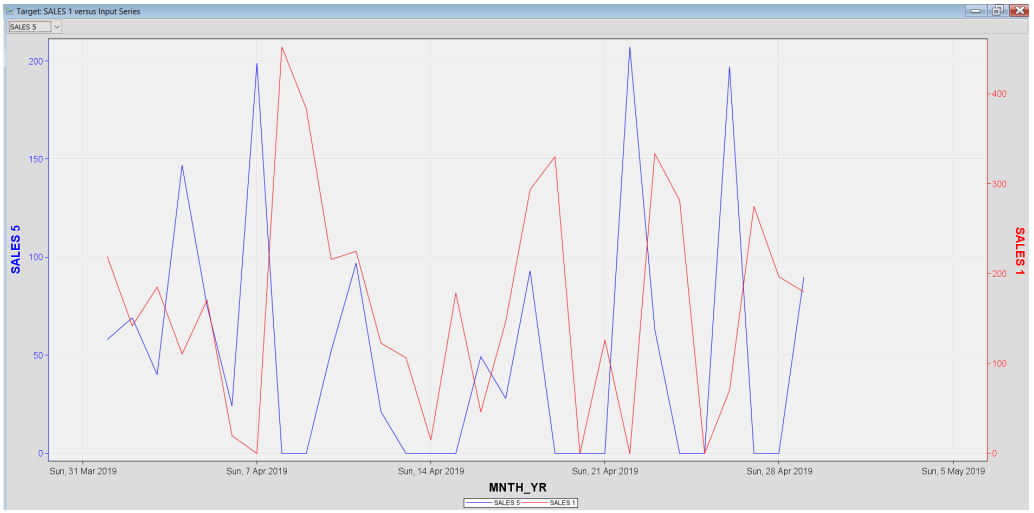


Diagram 2.2.21: Time Series Analysis of Sales1 vs Sales5

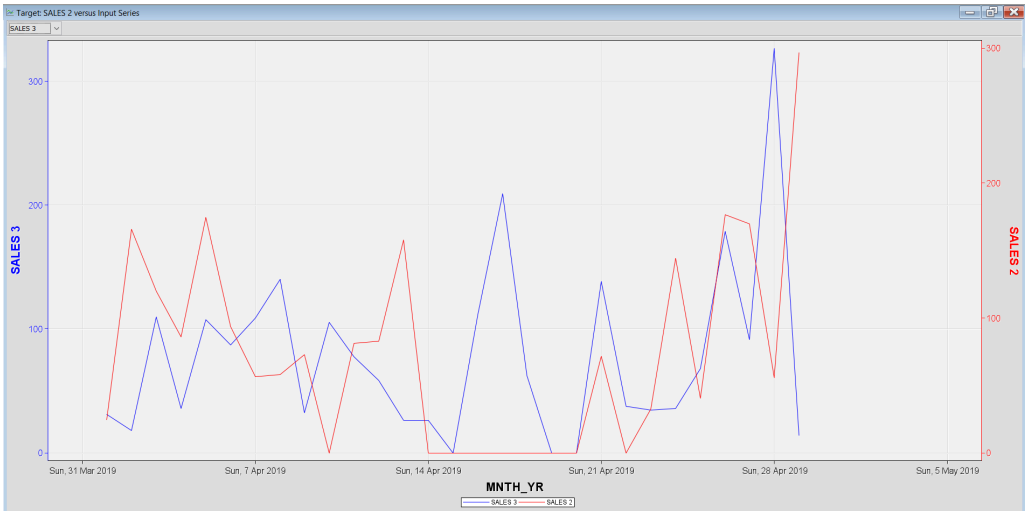


Diagram 2.2.22: Time Series Analysis of Sales2 vs Sales3

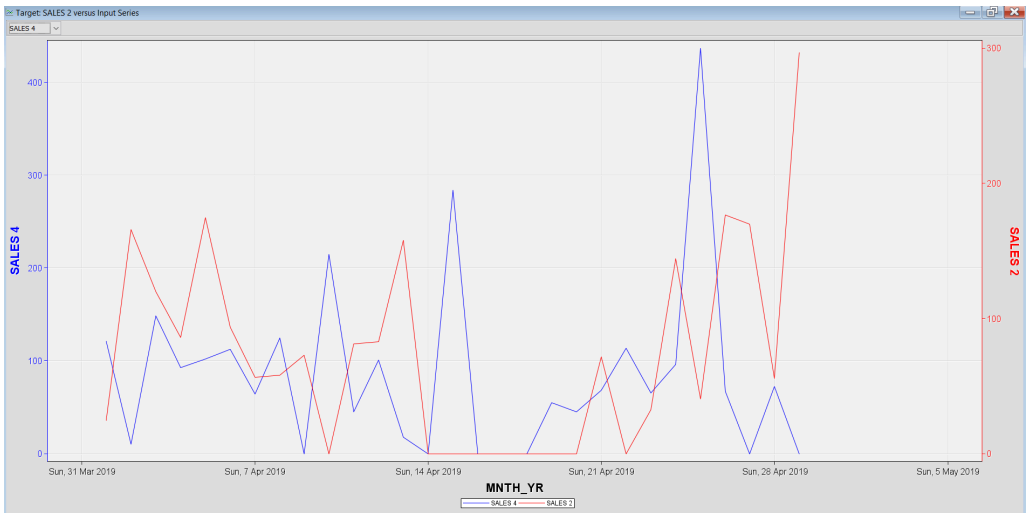


Diagram 2.2.23: Time Series Analysis of Sales2 vs Sales4

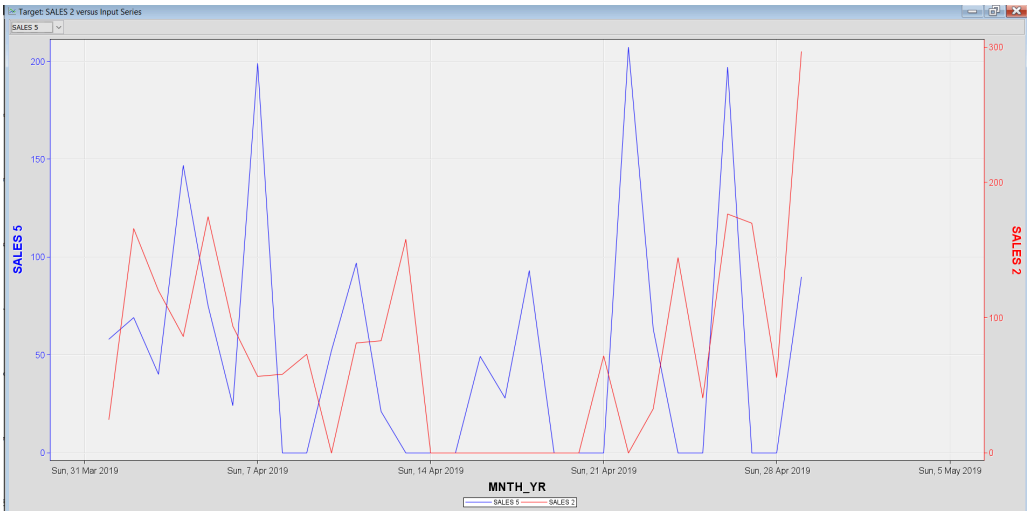


Diagram 2.2.24: Time Series Analysis of Sales2 vs Sales5

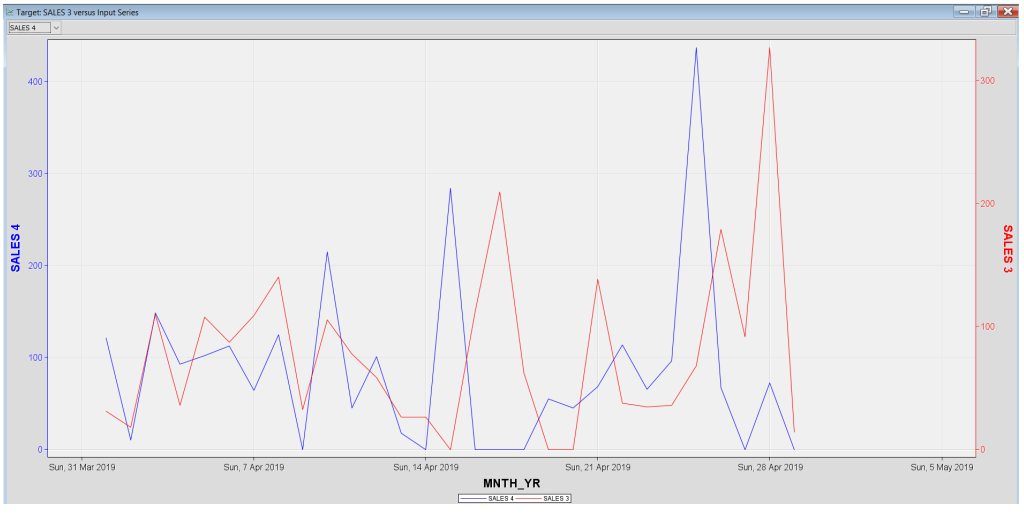


Diagram 2.2.25: Time Series Analysis of Sales3 vs Sales4

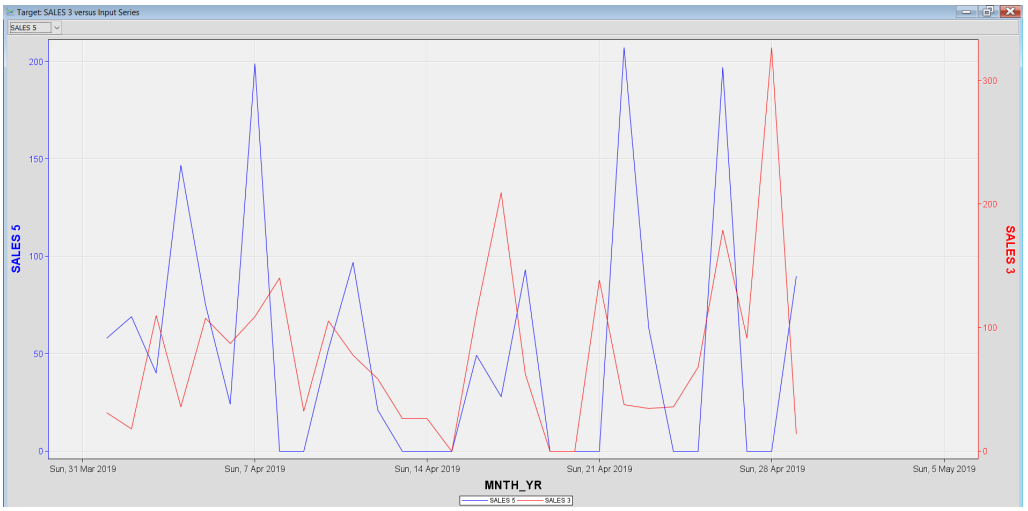


Diagram 2.2.26: Time Series Analysis of Sales3 vs Sales5

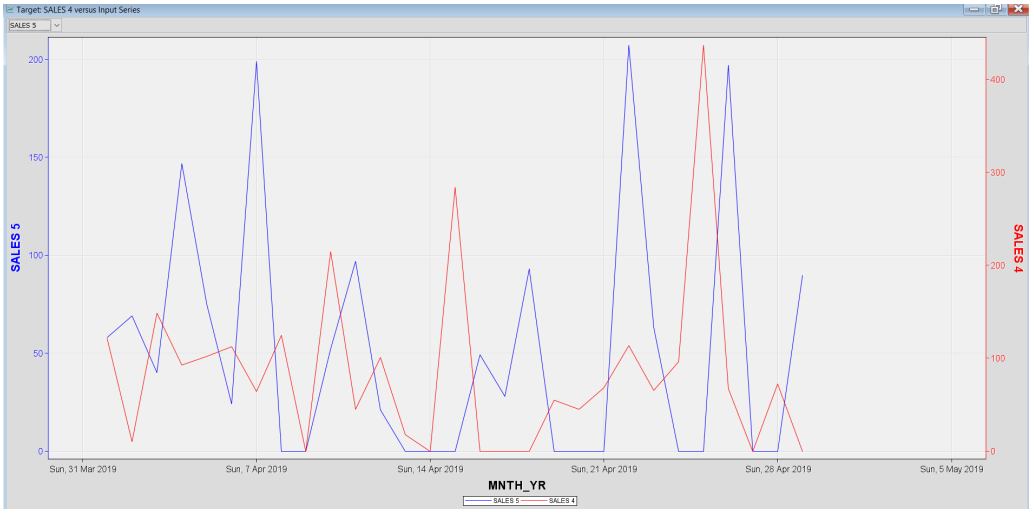


Diagram 2.2.27: Time Series Analysis of Sales4 vs Sales5

Through this insightful analysis, we can discern the distinct spending patterns of various generations by examining their sales volumes. Specifically, the data allows us to pinpoint the preferred spending days for each generation. Notably, on Sunday, April 21, 2018, we observe that Older Millennials, Gen Z, Gen X, and Baby Boomers collectively exhibit lower expenditure compared to their counterparts among the Younger Millennials. This nuanced understanding of generational spending habits unveils valuable insights for strategic decision-making and targeted marketing efforts.

2.3 Modify

The fundamental aim of data analysis is to thoroughly preprocess and convert unprocessed data in order to improve its integrity, relevance, and structure in preparation for

subsequent modeling endeavors. This procedure is important, as it considerably improves the precision and effectiveness of modeling efforts.

Data cleaning, error and inconsistency correction, and feature engineering, which includes the creation or modification of features to derive more meaningful information, are essential techniques that have been utilized. Furthermore, the processes of data scaling and normalization are of paramount importance as they serve to standardize numerical attributes, reduce discrepancies in scale, and guarantee a uniform distribution.

2.3.1 Number for missing values for each column:

→ **Tool: Talend Data Preparation**

Column	Data Type	Number of Missing Values
sales_id	Integer	0
transaction_id	Integer	0
transaction_date	Date	0
transaction_time	Time	0
customer_id	Integer	0
customer_since	Date	0
product_id	Integer	0
unit_price	Integer	0
product_group	String	0
product	String	0
current_retail_price	Integer	0
promo_yn	String	0
new_product_yn	String	0
customer_firstName	String	0
generation	String	0
gender	String	0
birth_year	Integer	0
quantity	Integer	14
instore_yn	String	3

Table 2.3.1: Number of missing values for each column

2.3.2 Visualizing Missing Data

This is a simple bar chart representing the number of missing values in each variable. It provides a quick overview of which variables have the most missing data.

→ Tool: SAS Enterprise Miner

- **Attributes: Quantity**

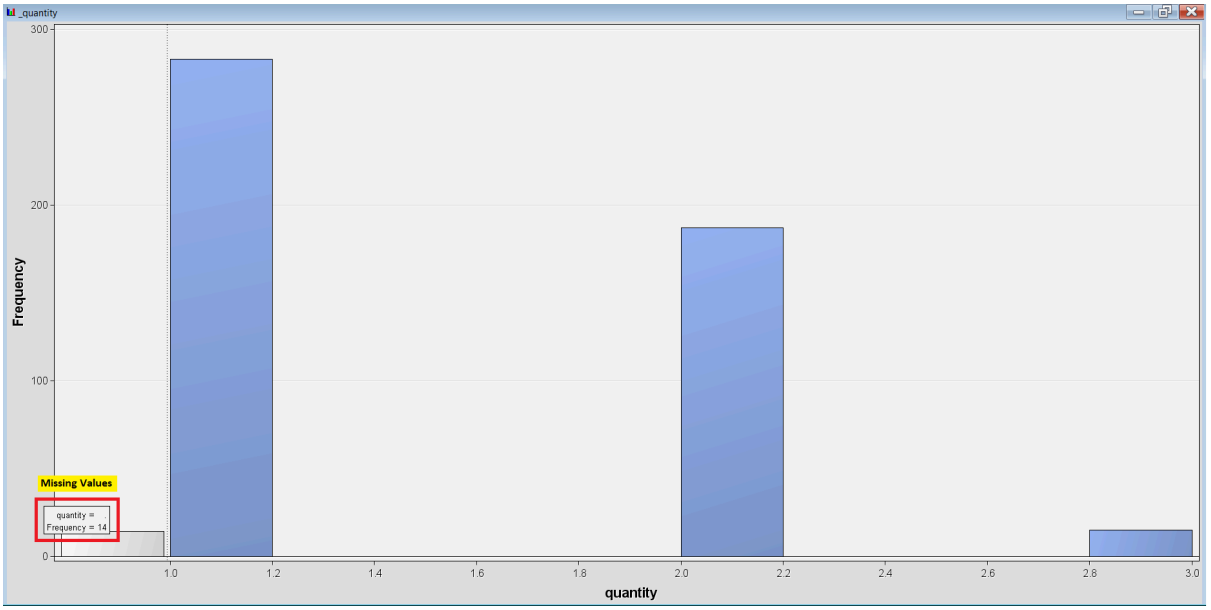


Diagram 2.3.1: Missing Data visualization for attribute 'Quantity'

- **Attributes: instore_yn**

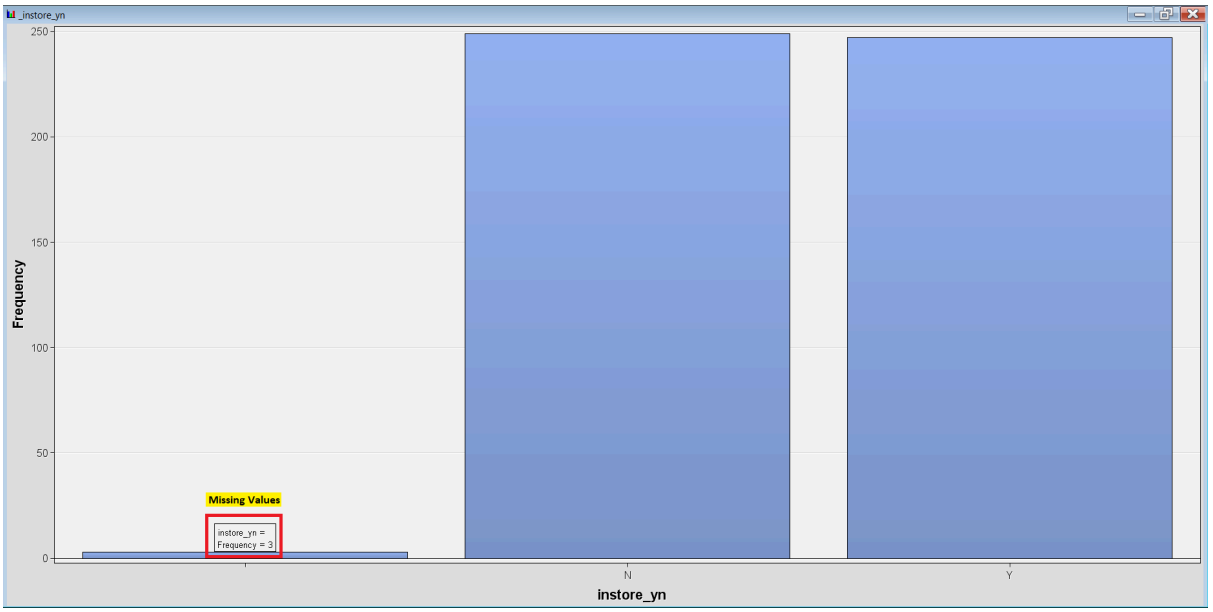


Diagram 2.3.2: Missing Data visualization for attribute 'instore_yn'

2.3.3 Impute Missing Data

"Impute Missing Data" describes how to estimate or add values to a dataset that are missing. In real-world datasets, missing data is a frequent occurrence that can result from a number of issues, including incomplete records, errors in data collection, and system failures. In order to create a more complete and useful dataset, imputation techniques are used to replace these missing values with estimated or predicted values.

Imputing missing data using the mean imputation method offers simplicity and ease of implementation, particularly suitable for continuous numeric variables like quantity while using the mode to fill in missing values for a categorical variable with binary values such as "yes" and "no" can help to preserve the overall distribution and maintain the balance between the two categories. This method maintains the original distribution by substituting the mean of observed values for missing values, thereby preserving the central tendency of the dataset.

→ **Tool: SAS Enterprise Miner**

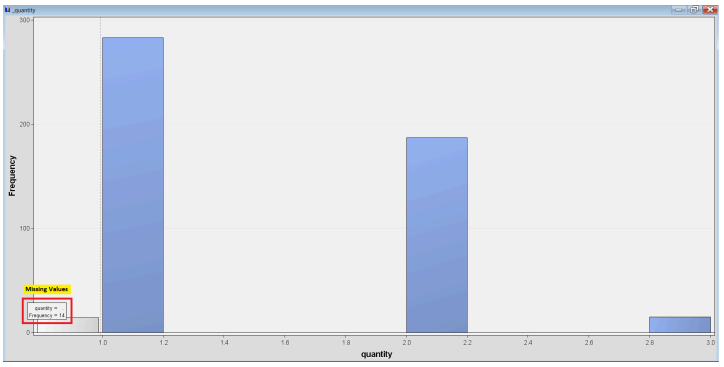
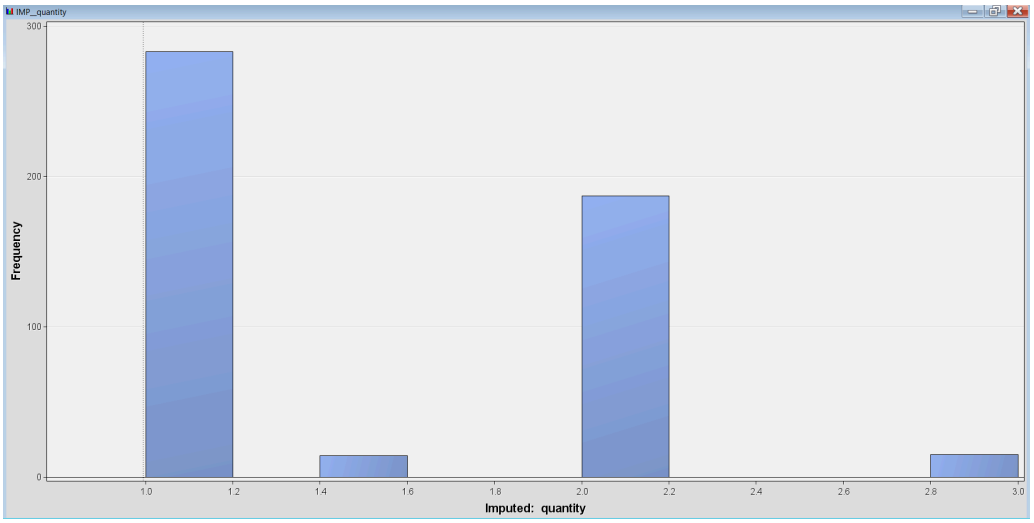
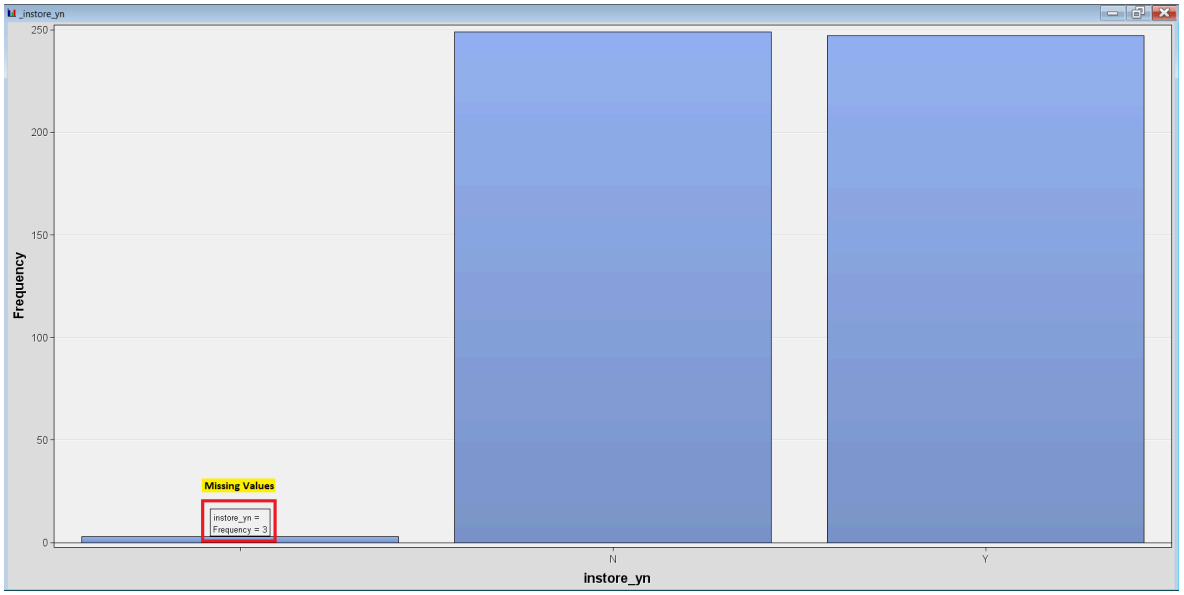
Variable: Quantity
Before

After


Diagram 2.3.3: Imputation for attribute 'Quantity'

Variable: instore_yn

Before



After

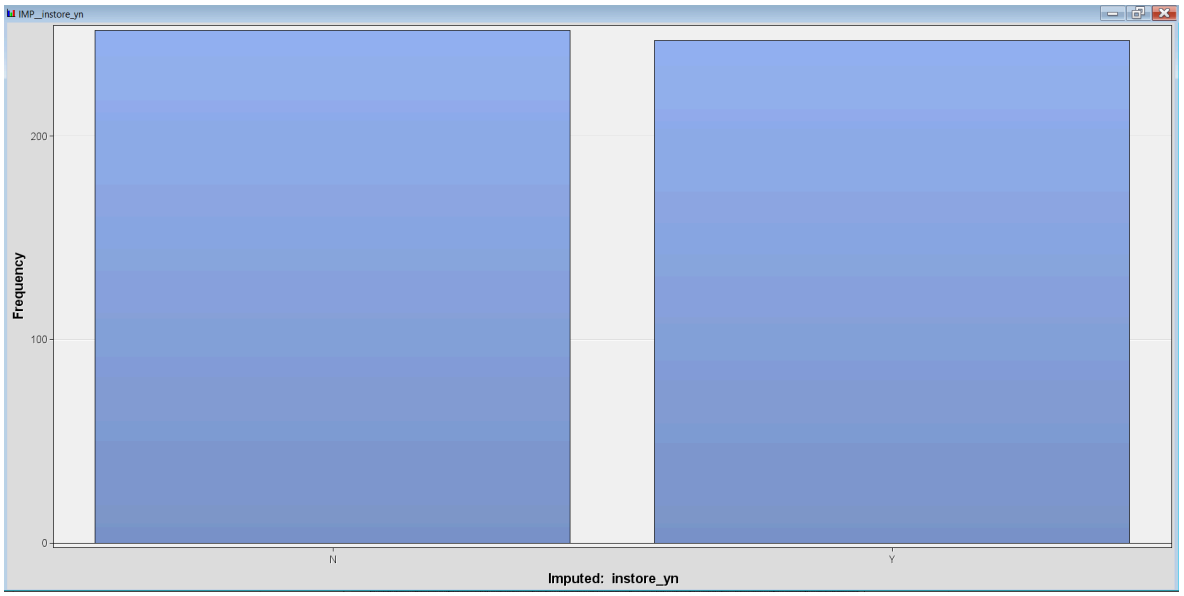


Diagram 2.3.4: Imputation for attribute 'instore_yn'

2.3.4 Filtering Outliers

Outliers, representing extreme or unusual values in a dataset, pose challenges in data analysis and machine learning. Since algorithms frequently need numerical inputs, handling outliers correctly is essential for making accurate decisions. Ensuring algorithms interpret data precisely through proper filtering and handling of outliers increases the importance and dependability of machine learning tasks.

→ Tool: SAS Enterprise Miner

Following is the diagram to filter the outlier.

Diagram 2.3.5: Nodes involved in FIltering Outliers

The dataset undergoes filtration using the 'Filter' node, employing the 'Standard Deviations from the Mean' method in order to filter the outliers. The filtering results are presented below.

Filter Limits for Interval Variables
(maximum 500 observations printed)

Variable	Role	Minimum	Maximum	Filter Method	Keep Missing Values	Label
birth_year	INPUT	1928.30	2021.54	STDDEV	Y	
current_retail_price	INPUT	-5.44	12.92	STDDEV	Y	
customer_COUNT_sales_	INPUT	-11.14	20.95	STDDEV	Y	customer.COUNT(sales)
customer_MEAN_sales__IMP__quanti	INPUT	0.27	2.62	STDDEV	Y	customer.MEAN(sales. IMP__quanti
customer_MEAN_sales_unit_price_	INPUT	-2.15	9.28	STDDEV	Y	customer.MEAN(sales.unit_price)
customer_SUM_sales__IMP__quantit	INPUT	-13.69	27.25	STDDEV	Y	customer.SUM(sales. IMP__quantit
customer_SUM_sales_unit_price_	INPUT	-52.77	89.42	STDDEV	Y	customer.SUM(sales.unit_price)
product_COUNT_sales_	INPUT	-1.21	20.28	STDDEV	Y	product.COUNT(sales)
product_SUM_sales__IMP__quantity	INPUT	-4.45	32.78	STDDEV	Y	product.SUM(sales. IMP__quantity
product_SUM_sales_unit_price__sa	INPUT	-9.24	68.55	STDDEV	Y	product.SUM(sales.unit_price)(sa
quantity	INPUT	-0.20	3.09	STDDEV	Y	
unit_price	INPUT	-5.64	12.77	STDDEV	Y	

Excluded Class Values
(maximum 500 observations printed)

Variable	Role	Level	Train Count	Train Percent	Label	Filter Method
product_group	INPUT	MERCHANDI	4	0.80160		MINPCT

Number Of Observations

Data Role	Filtered	Excluded	DATA
TRAIN	447	52	499

Diagram 2.3.6: Results after FIltering Outliers

2.3.6 Encoding Categorical Variables

Categorical variables, which denote discrete groups or categories, are frequently challenging in data analysis and machine learning because algorithms usually need numerical inputs. In order to ensure that algorithms can interpret and use categorical variables in decision-making processes with accuracy and significance, it is essential that these variables be encoded correctly in machine learning tasks.

→ Tool: Talend Data Preparation

Following tables are the result of before and after we encode the categorical variables.

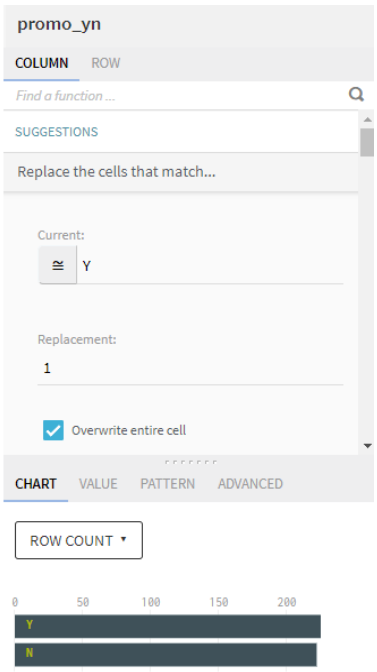
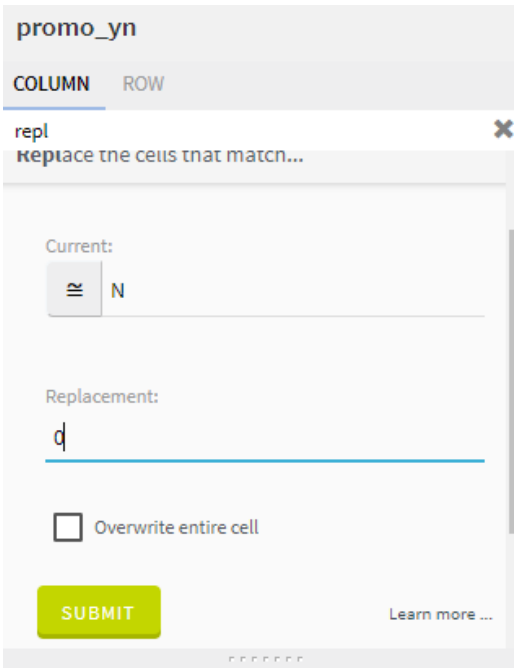
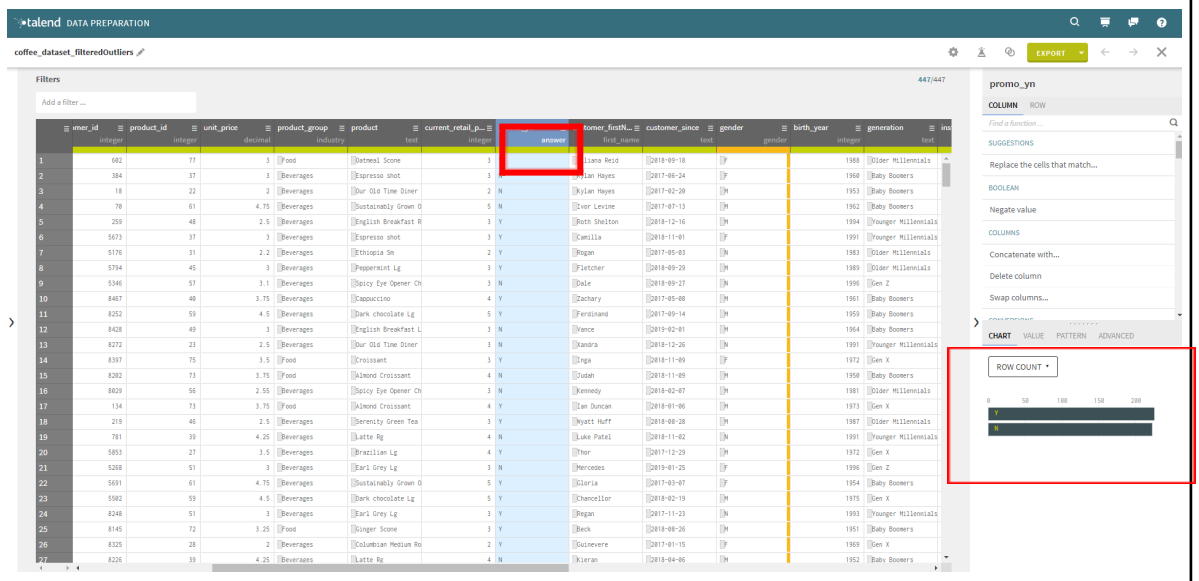
Before	After
	

Diagram 2.3.7: Steps in Encoding Categorical variables

Variable involved:
1) promo_yn, transaction_date, birthyear current_retail_price, promo_yn, new_product_yn, quantity, instore_yn, customer_COUNT_sales, customer_MEAN_sales_IMP_quantity, customer_MEAN_sales_unit_price, customer_MODE_sales_IMP_instore_yn

Variable: promo_yn

Before



After

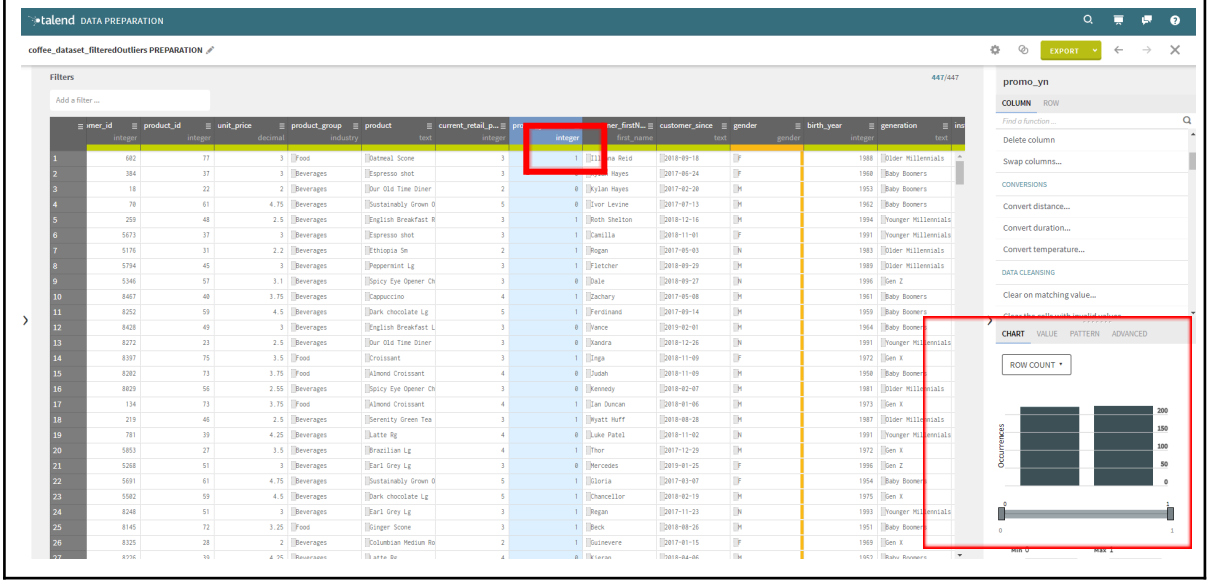
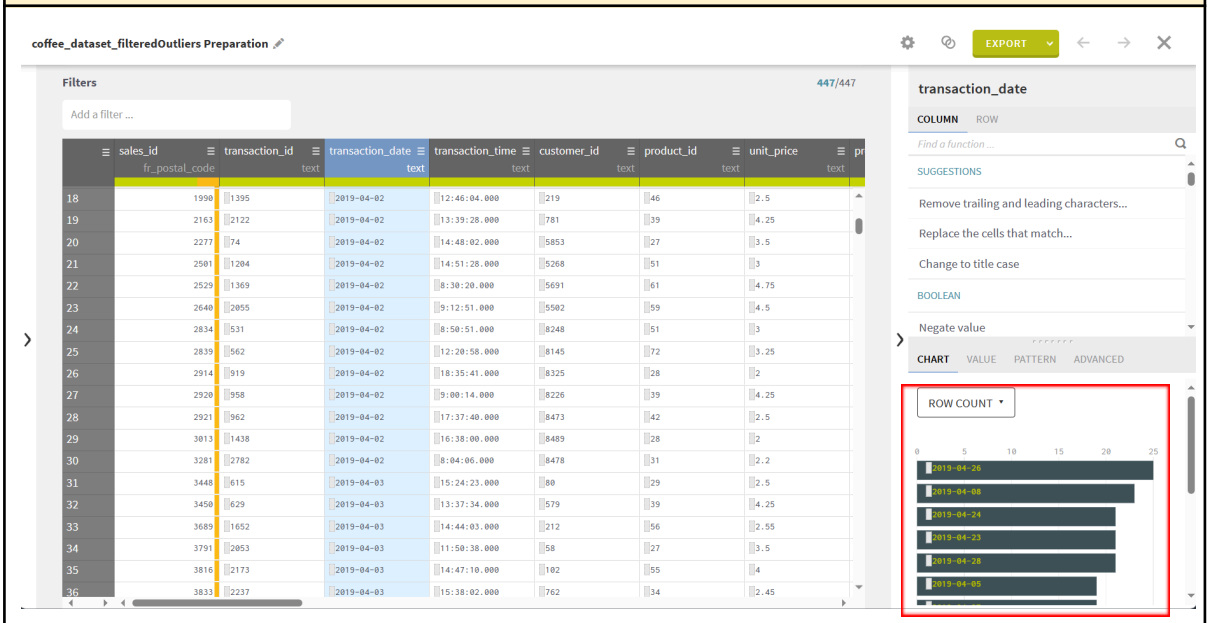


Diagram 2.3.8: Results for Encoding variable 'promo_yn'

Variable: transaction_date

Before



After

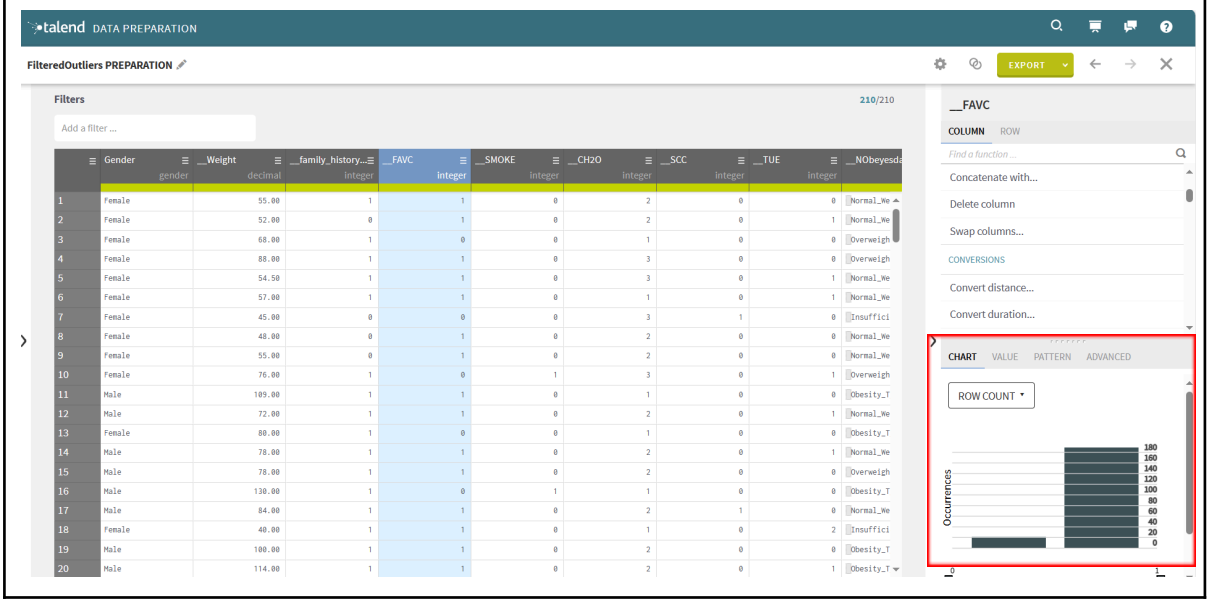
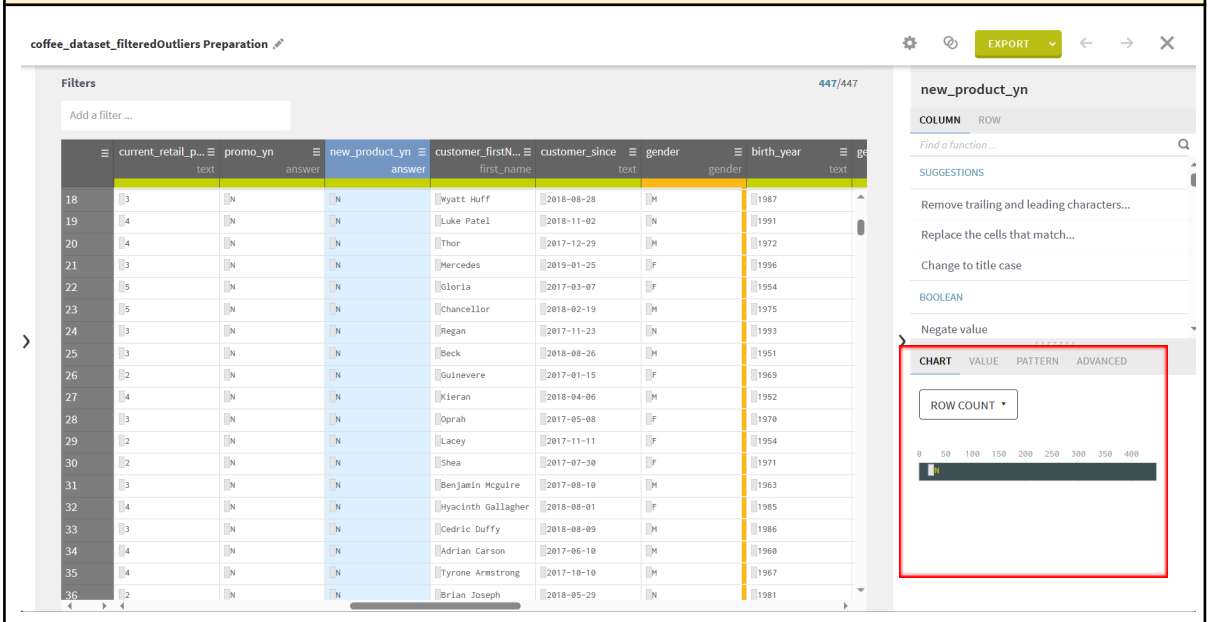


Diagram 2.3.9: Results for Encoding variable 'transaction_date'

Variable: new_product_yn

Before



After

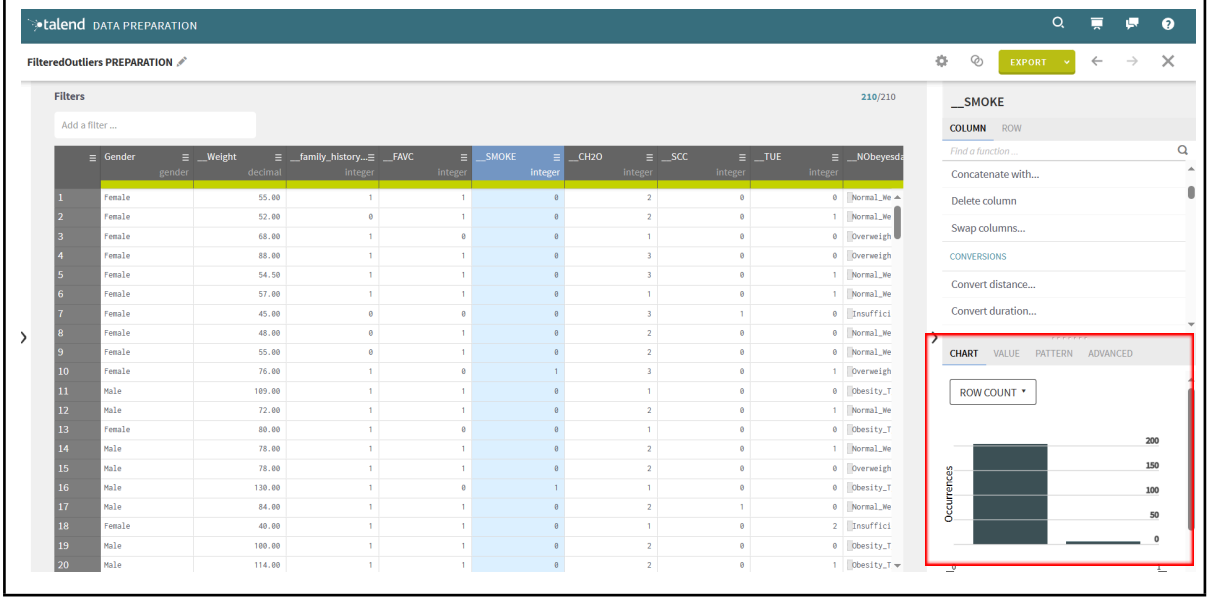
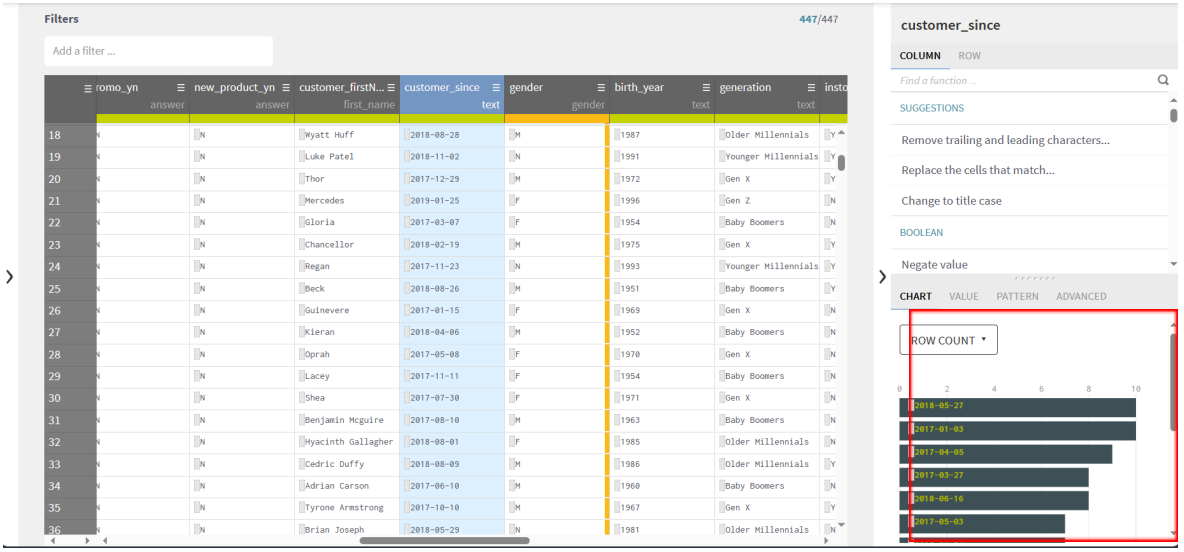


Diagram 2.3.10: Results for Encoding variable 'new_product_yn'

Variable: customer_since

Before



After

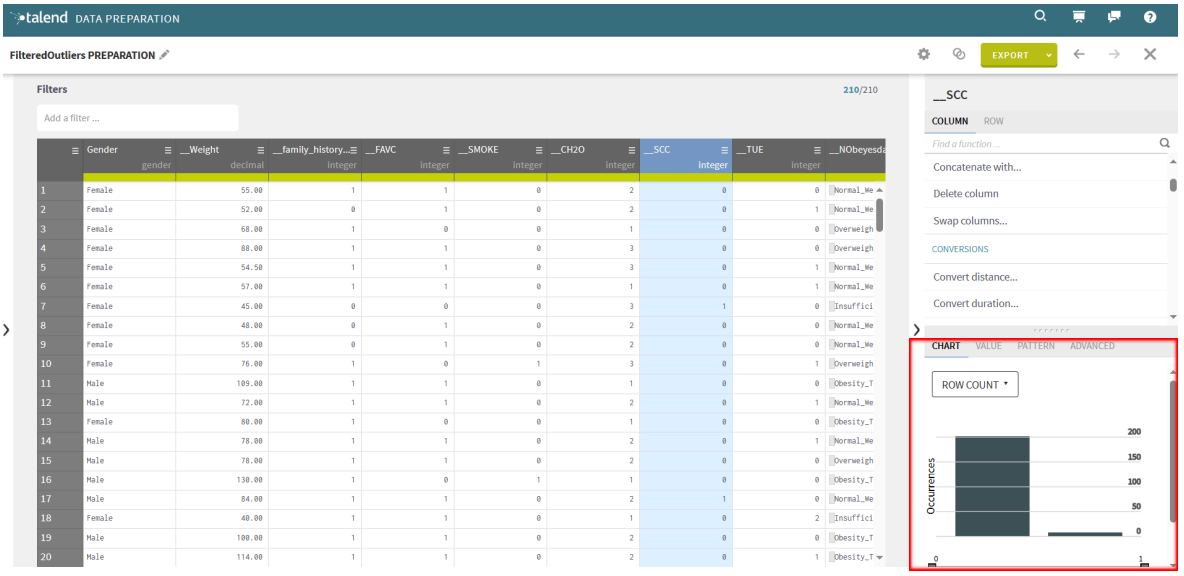
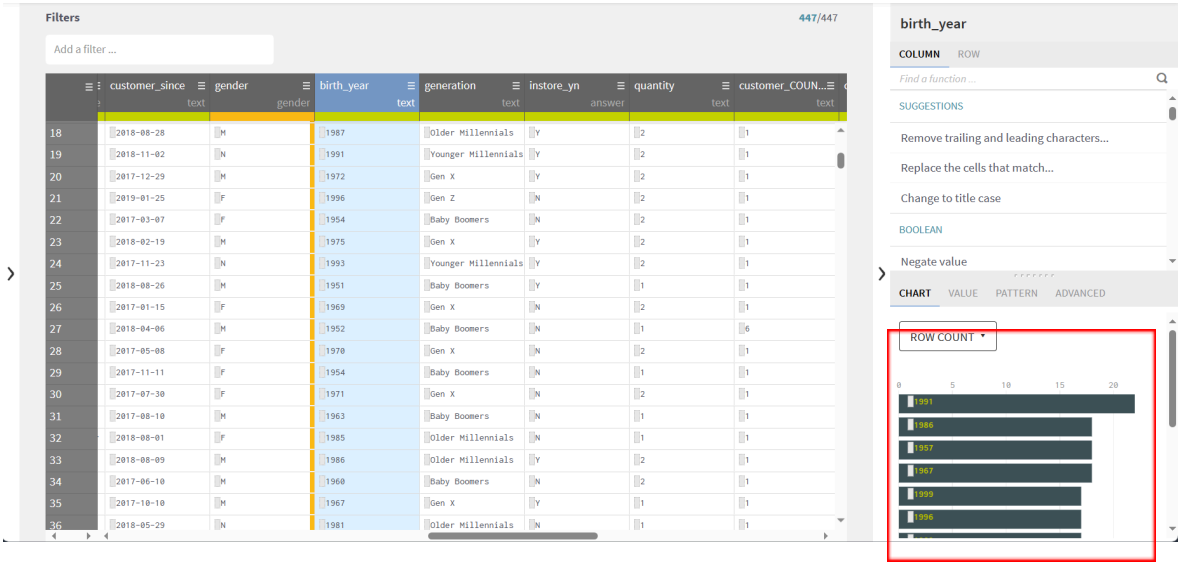


Diagram 2.3.11: Results for Encoding variable 'customer_since'

Variable: birth_year

Before



After

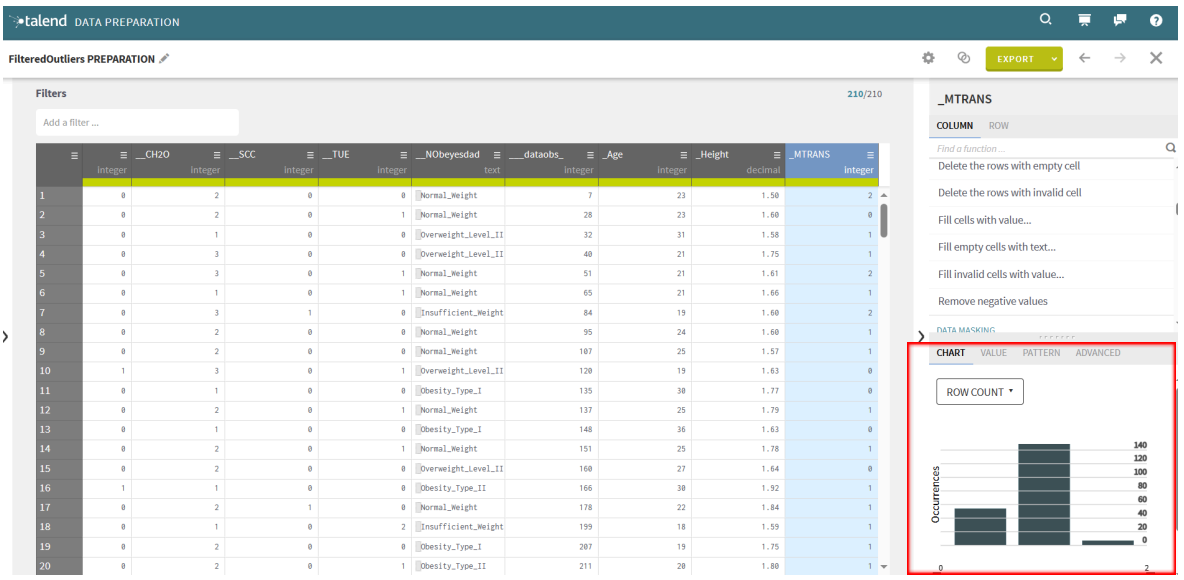
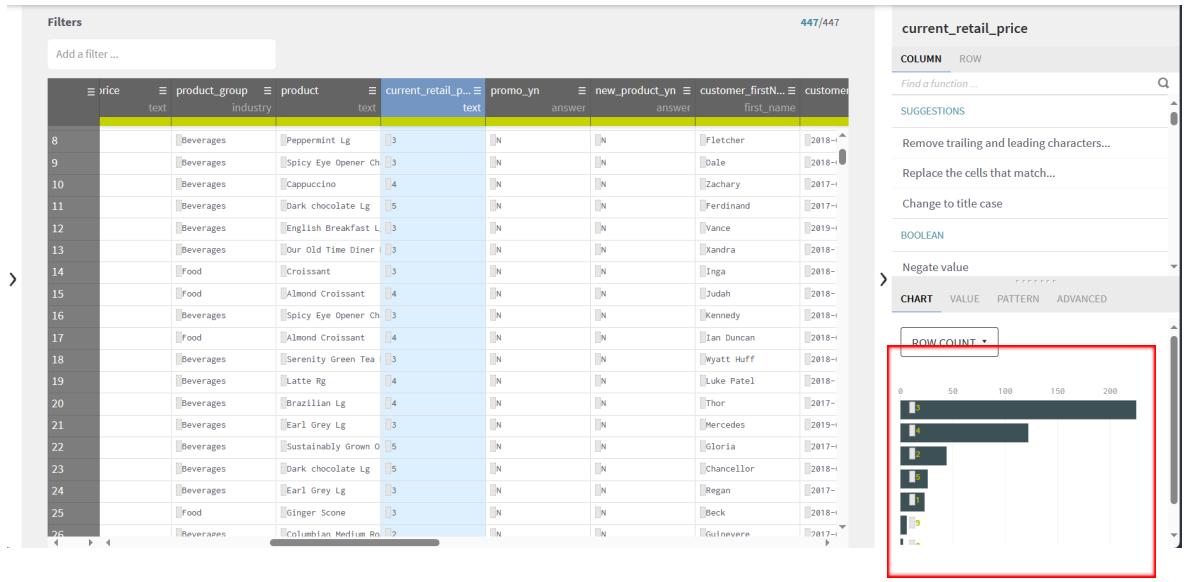


Diagram 2.3.12: Results for Encoding variable 'birth_year'

Variable: current_retail_price

Before



After

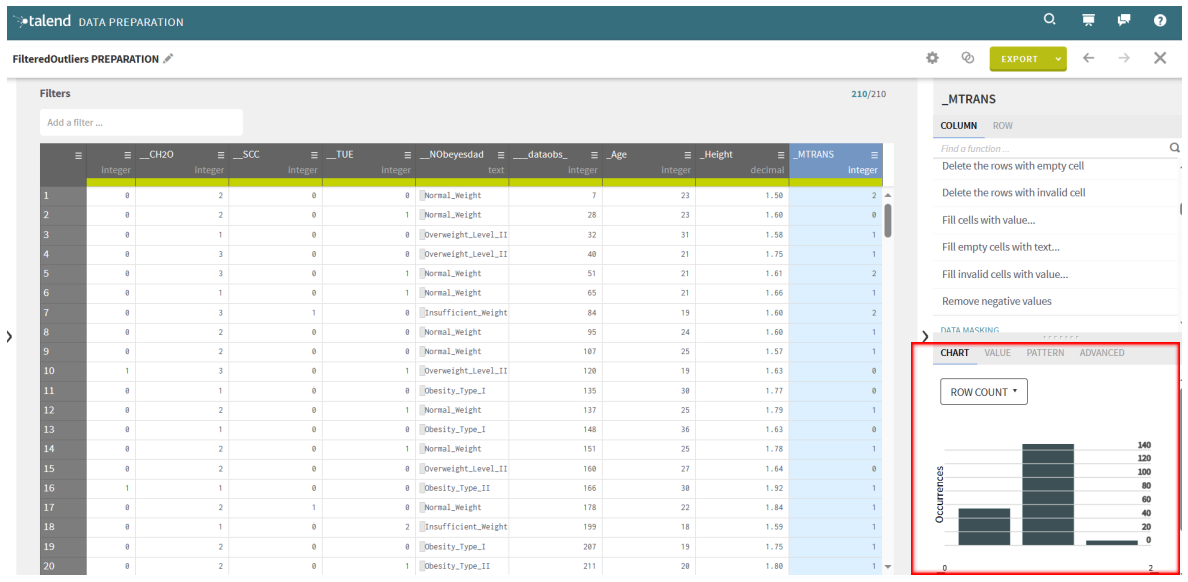
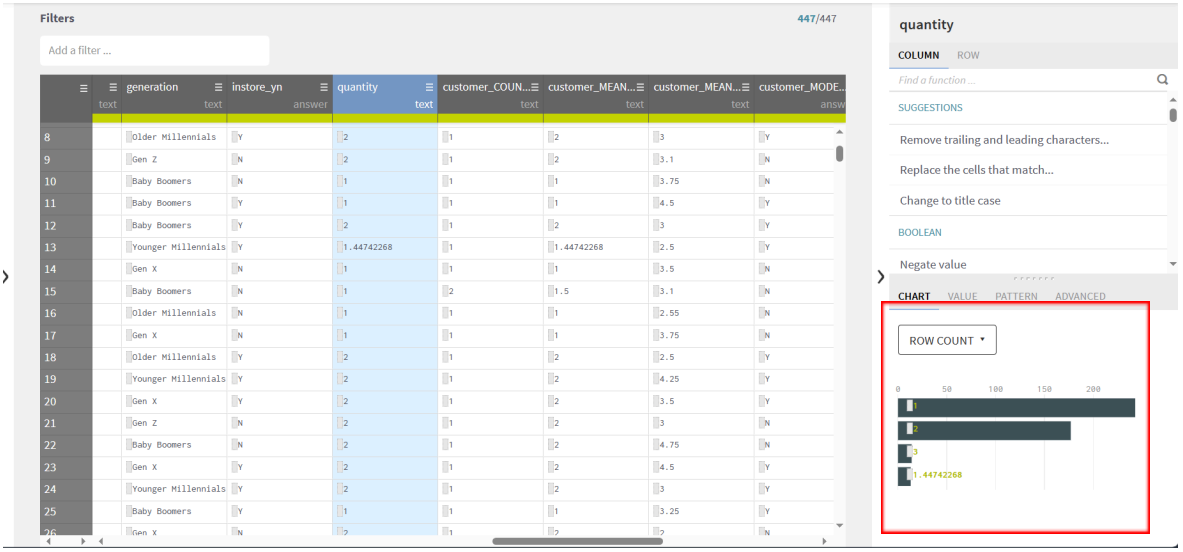


Diagram 2.3.13: Results for Encoding variable 'current_retail_price'

Variable: quantity

Before



After

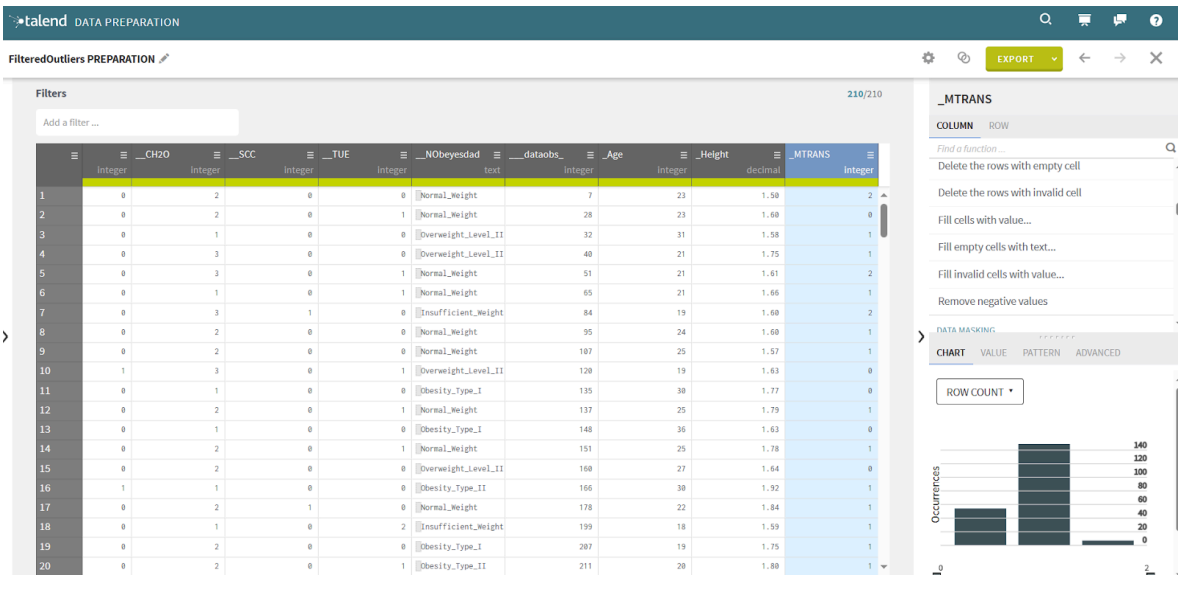
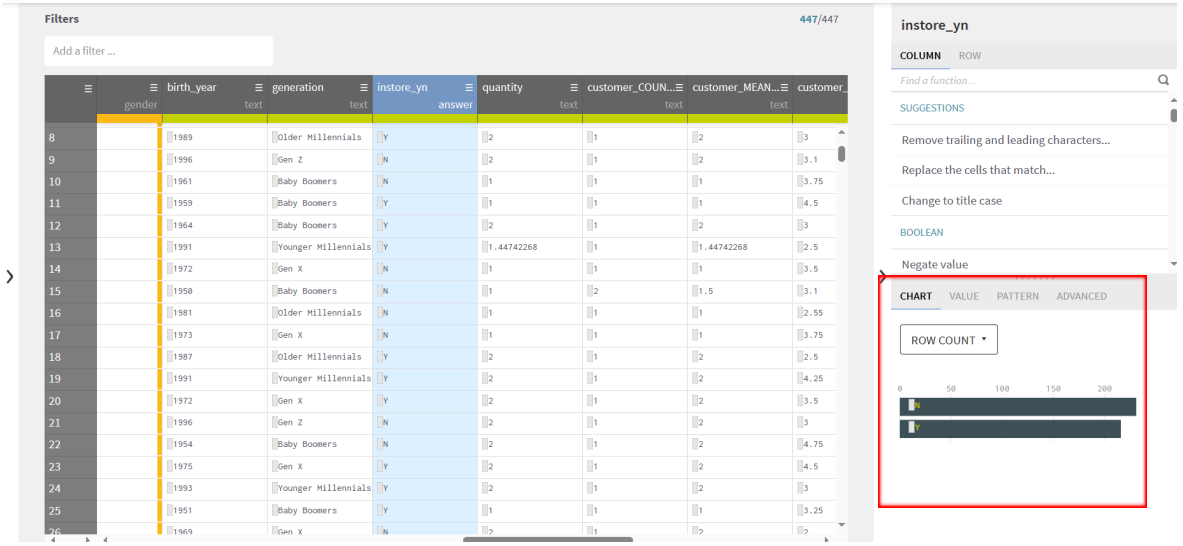


Diagram 2.3.14: Results for Encoding variable 'quantity'

Variable: instore_yn

Before



After

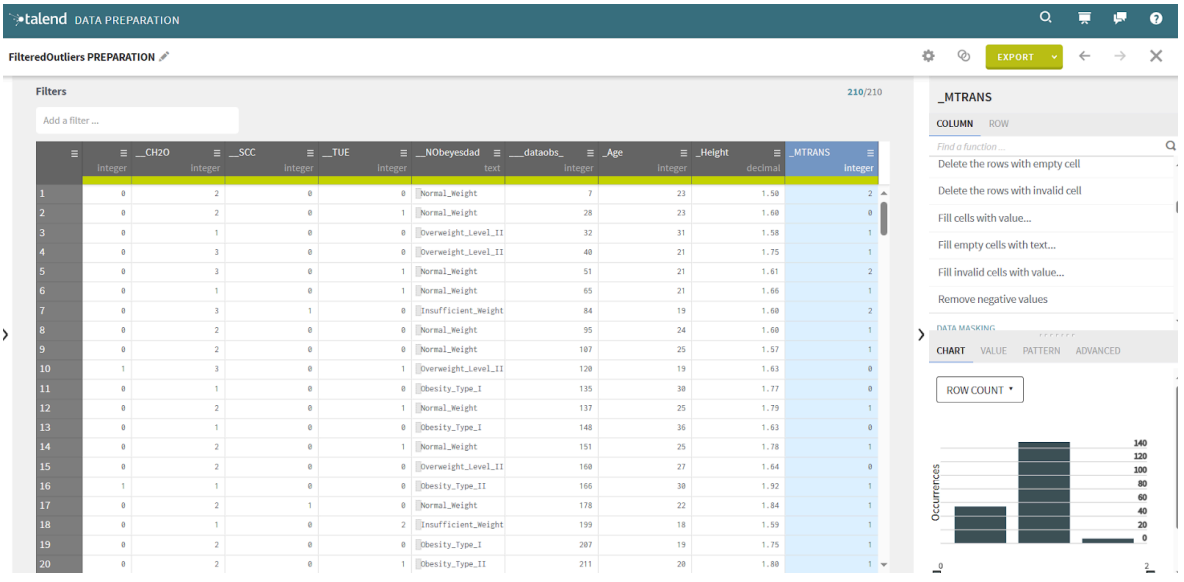


Diagram 2.3.15: Results for Encoding variable 'instore_yn'

2.3.7 Standardize Data Types and Format

In the context of data preparation and analysis, standardizing data types and formats is a critical step to ensure consistency, accuracy, and reliability. This process involves transforming diverse attributes in which variables are formatted consistently and conform to standardized data types like dates, decimals, or integers. Standardized data types and formats contribute to better data quality, making it easier to identify and address issues like missing values or outliers.

→ **Tool: Talend Data Preparation**

1. When first exporting the sampling output in data prep, most data has a white space in front, thus we use RegEx on each column to remove the string.

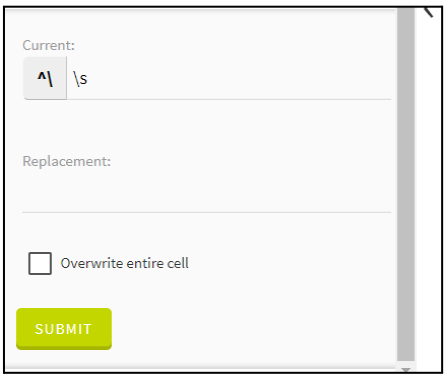
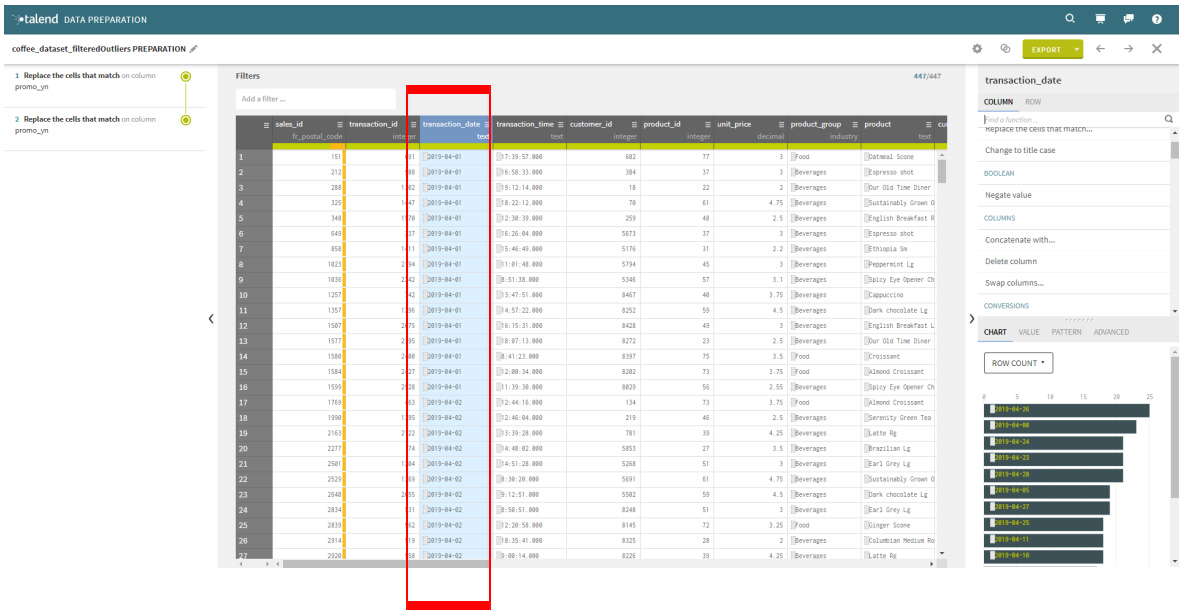


Diagram 2.3.16: Removing white spaces in column name using RegEx

Before



After

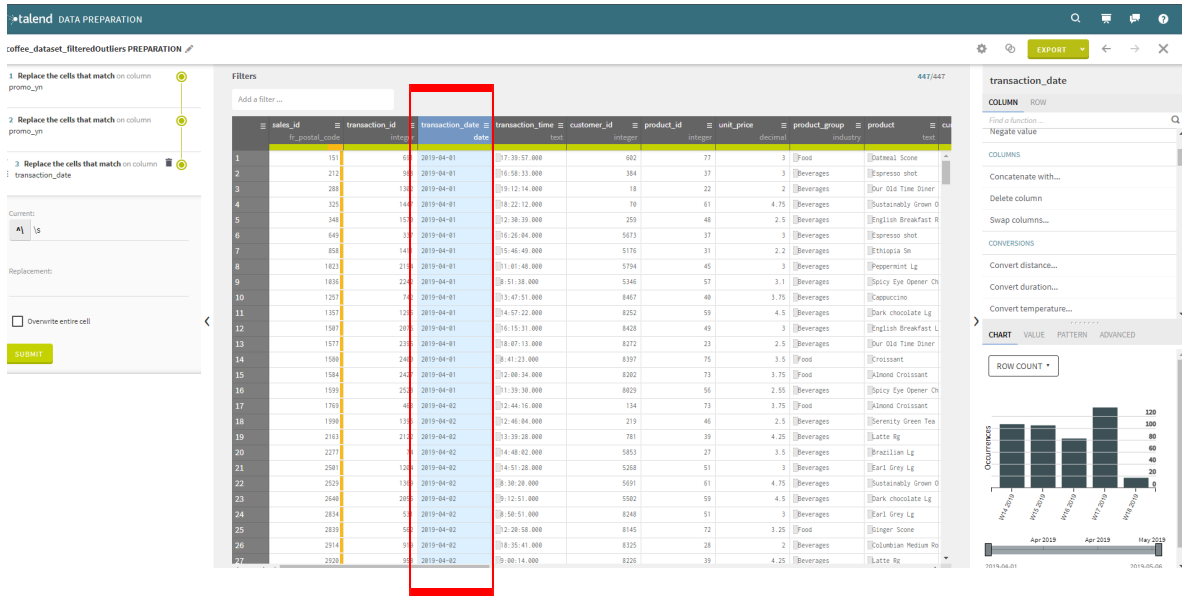


Diagram 2.3.17: Results for removing white spaces in column name

2. We standardized the data type for each column based on their suitable data type. By clicking on ‘Replace cells that match’, click on a regular expression and do `\..*` to remove the decimal points and remain the integers.

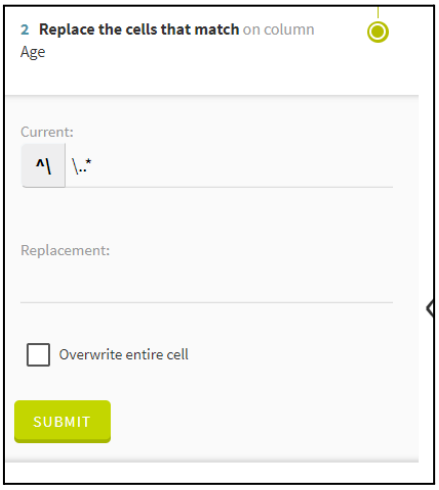


Diagram 2.3.18: Removing decimal points for columns

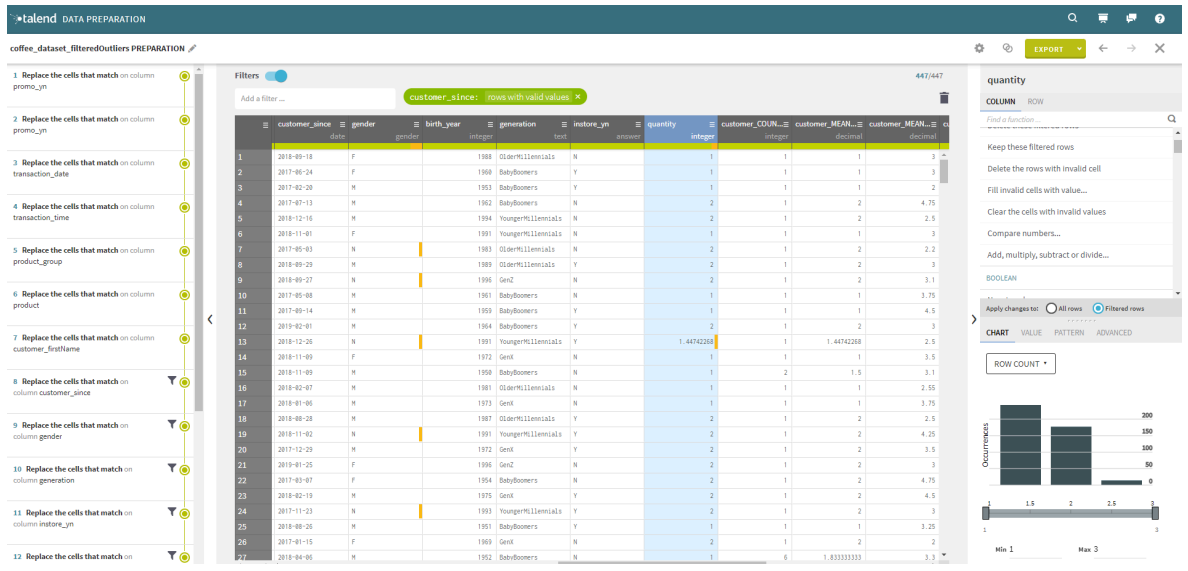
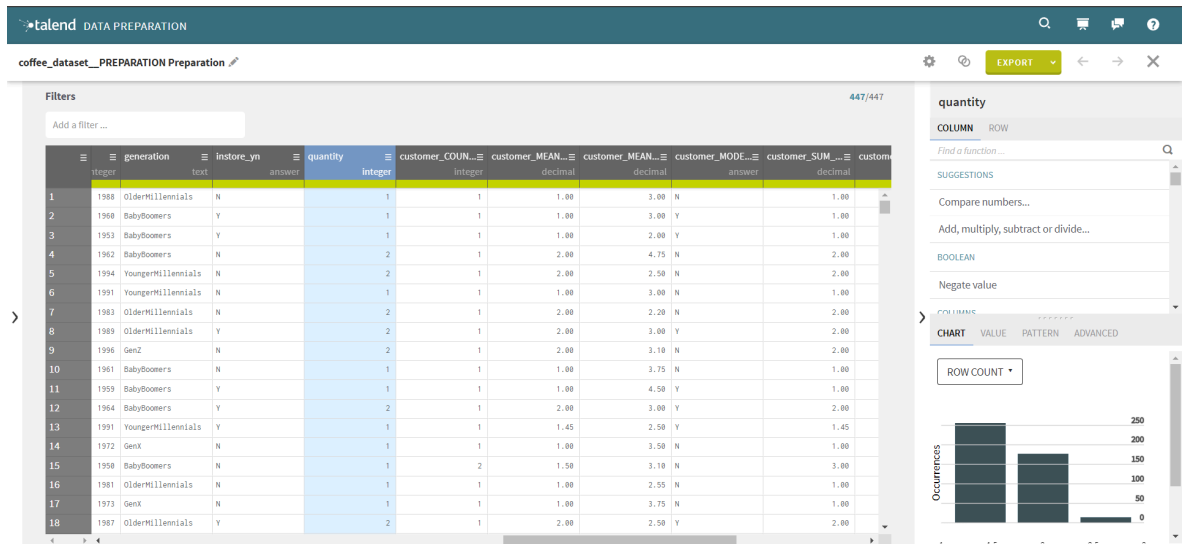
quantity
Before

After


Diagram 2.3.19: Results for removing decimal points for column ‘quantity’

3. For decimal points, we round values using ceil mode with the precision of 2.

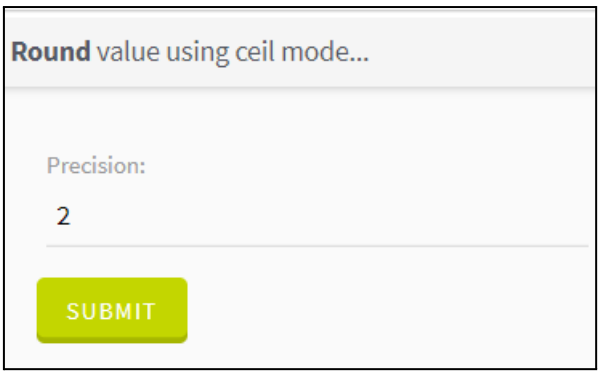
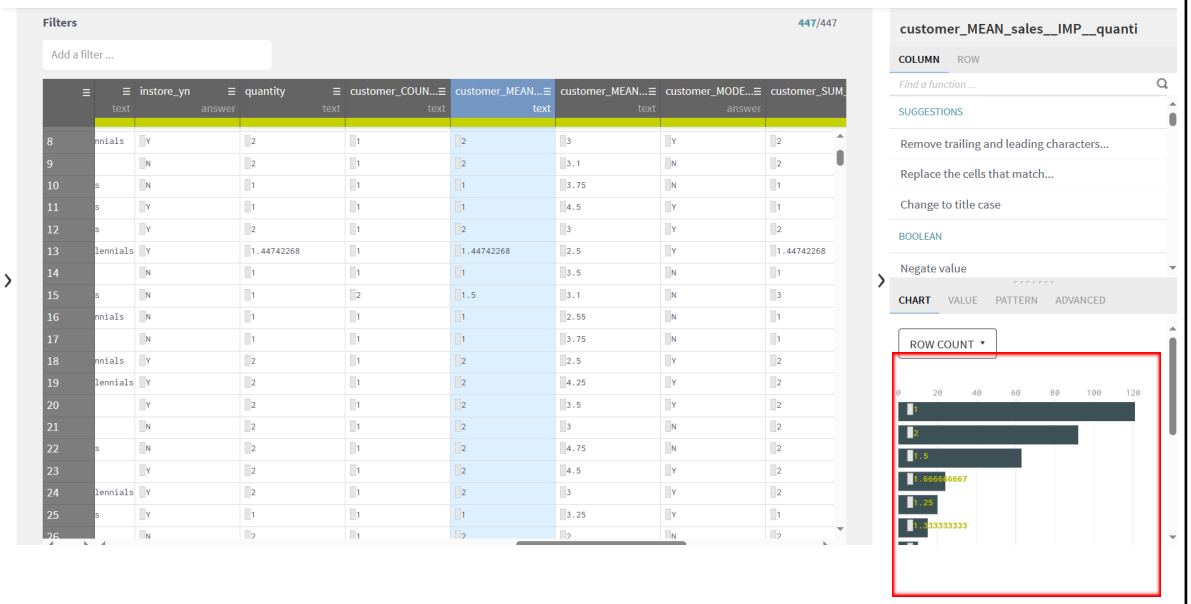


Diagram 2.3.20: Round decimal points using ceil mode

customer_MEAN_sales__IMP__quantity

Before

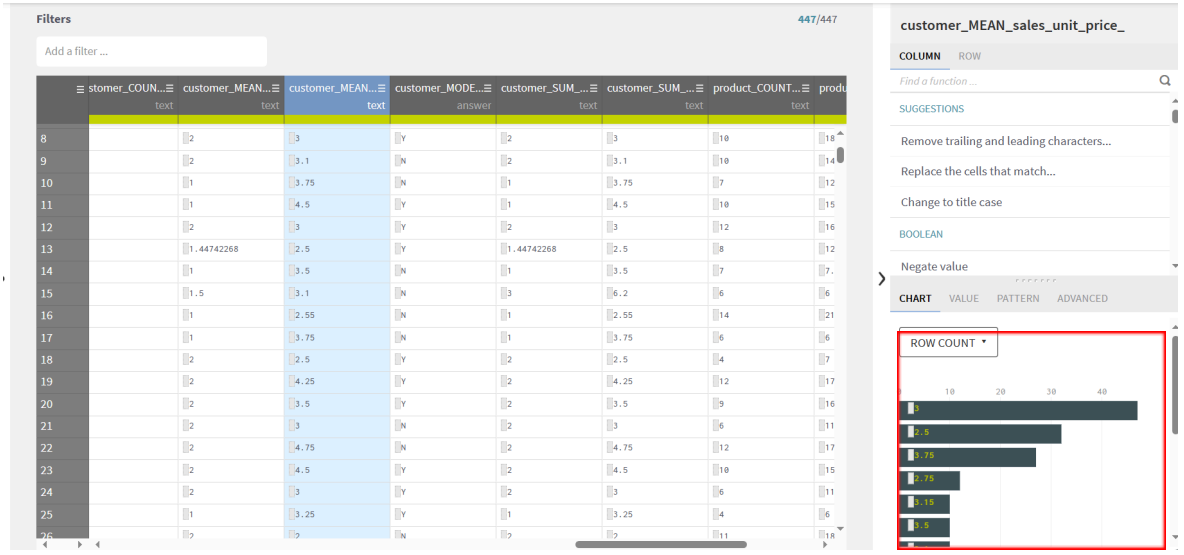


After

Diagram 2.3.21: Results for rounding decimal points for 'customer_MEAN_sales__IMP__quantity'

customer_MEAN_sales_unit_price_

Before



After

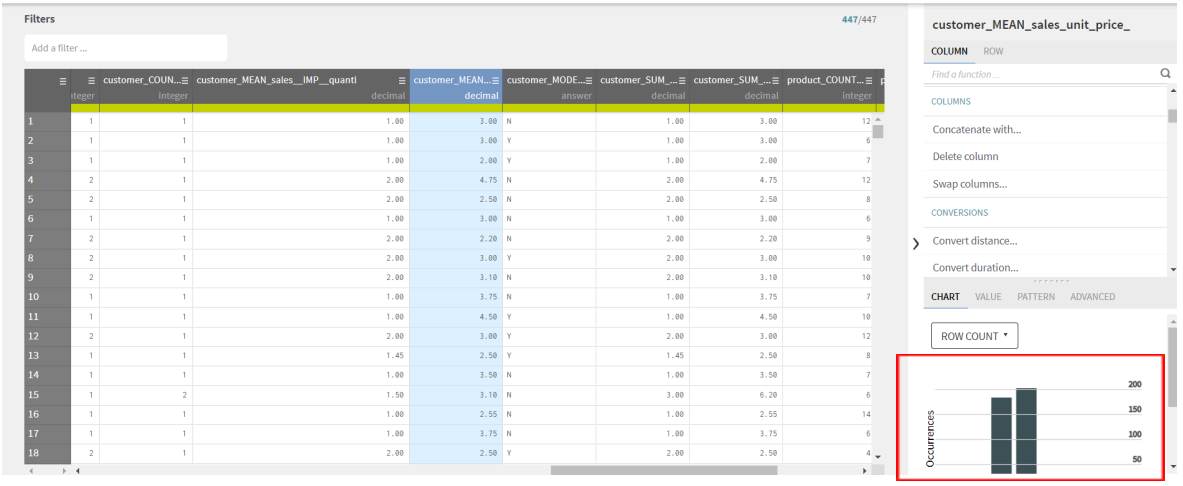
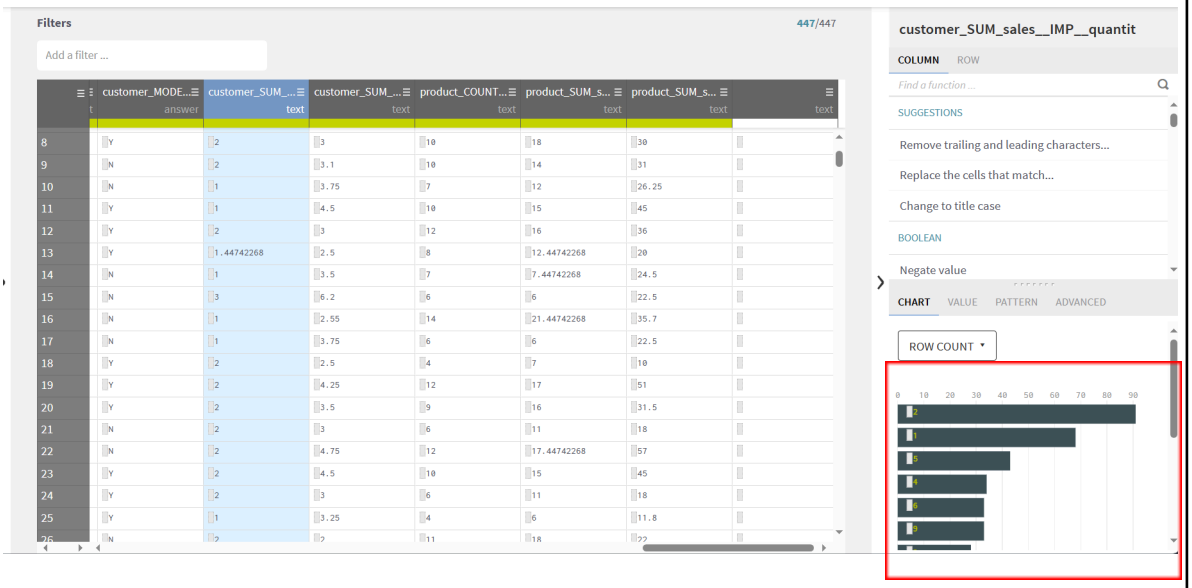


Diagram 2.3.22: Results for rounding decimal points for 'customer_MEAN_sales_unit_price_'

customer_SUM_sales__IMP__quantity

Before



After

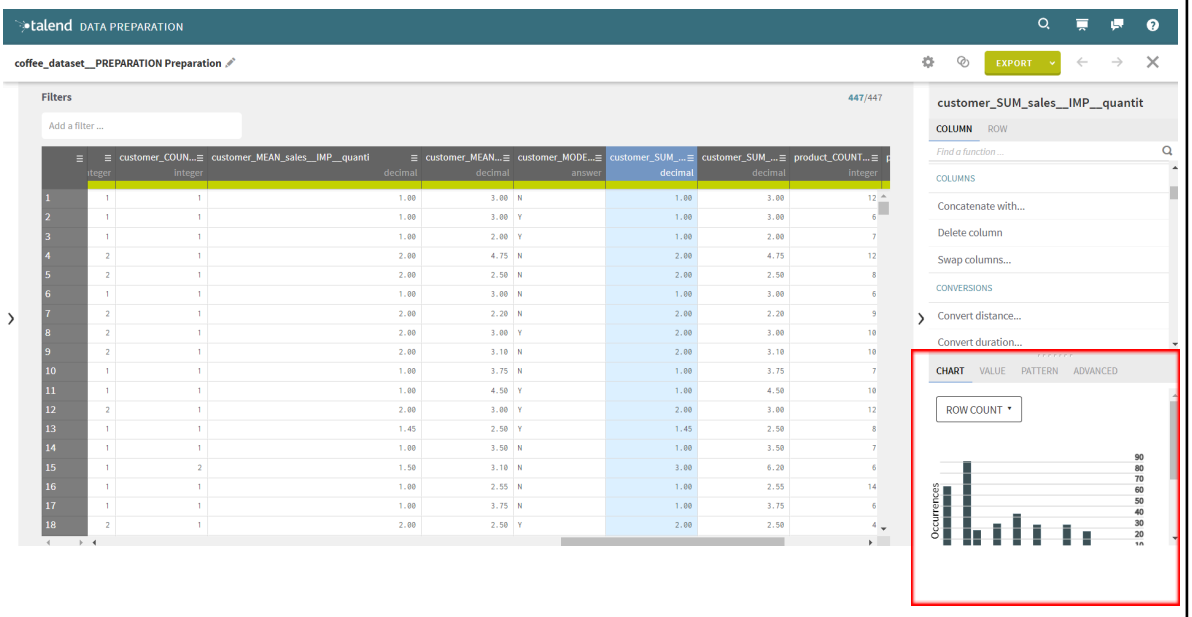
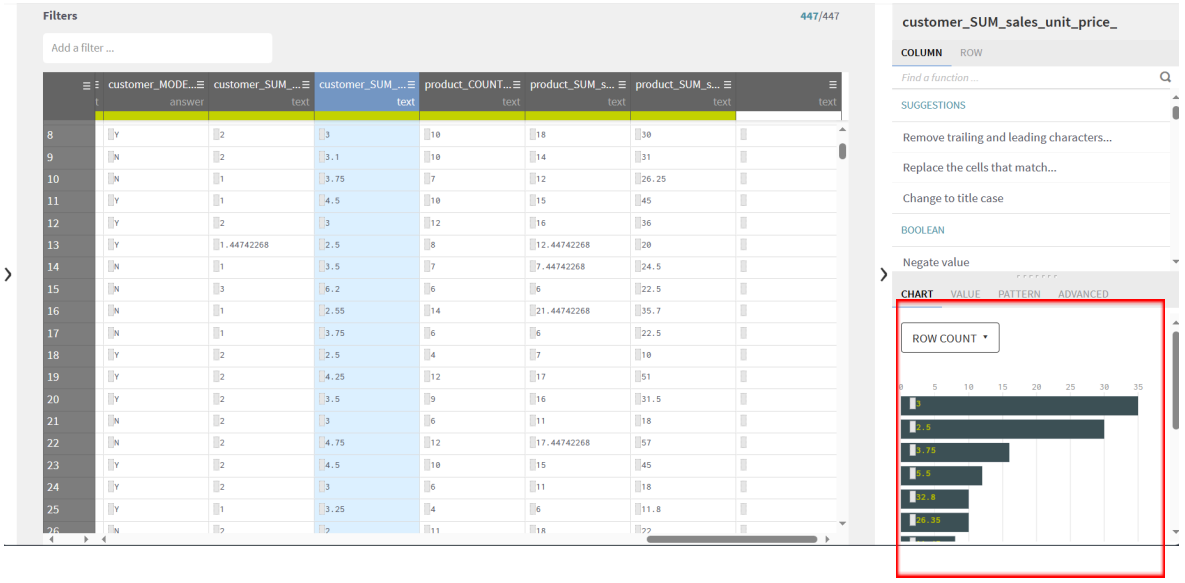


Diagram 2.3.23: Results for rounding decimal points for 'customer_SUM_sales__IMP__quantity'

customer_SUM_sales_unit_price_

Before



After

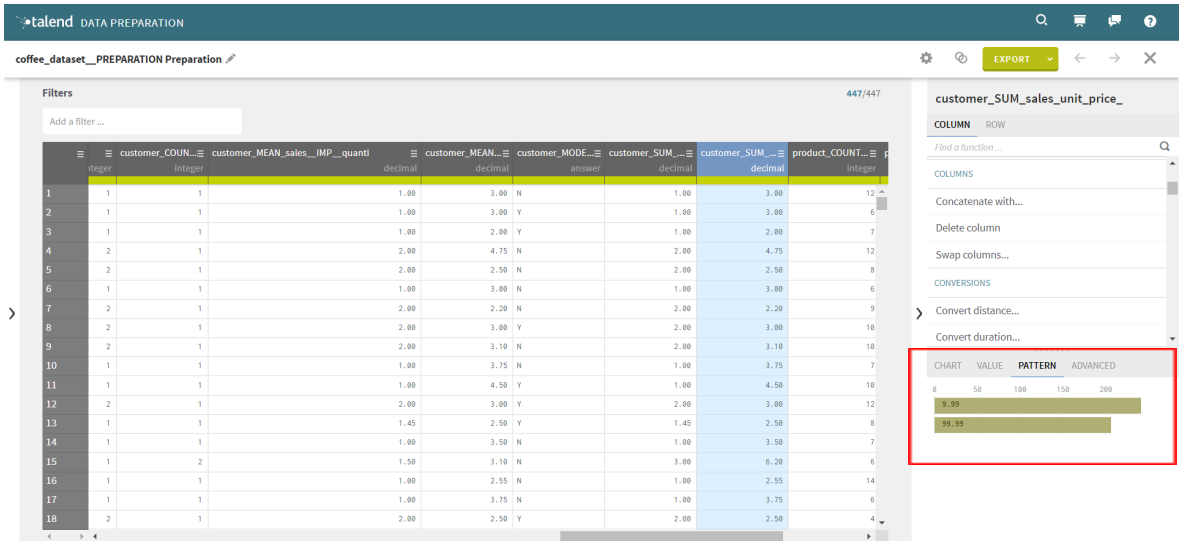
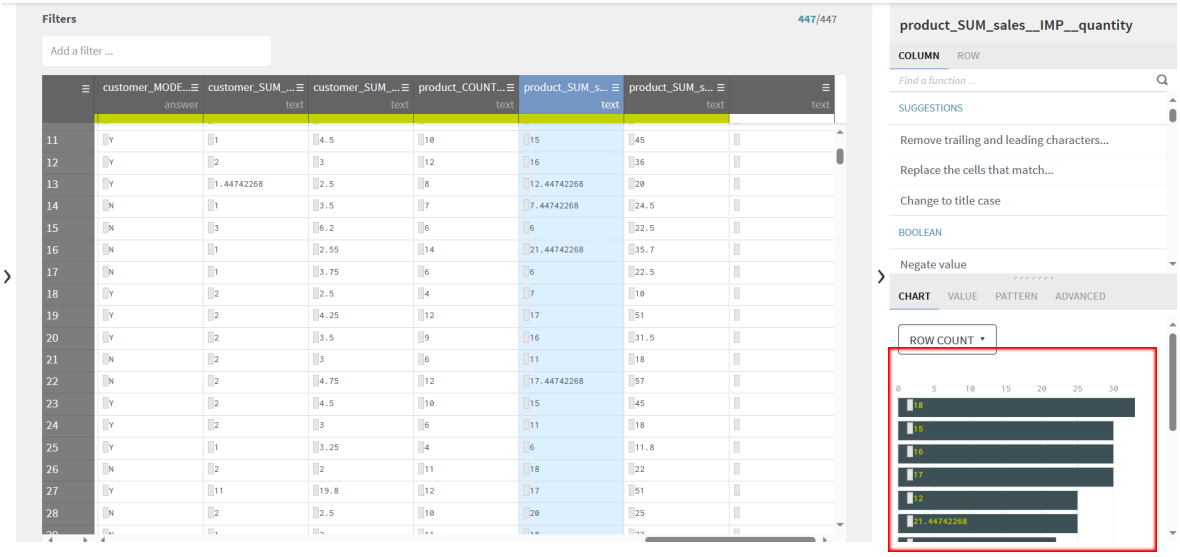


Diagram 2.3.24: Results for rounding decimal points for ‘ customer_SUM_sales_unit_price_’

product_SUM_sales__IMP__quantity

Before



After

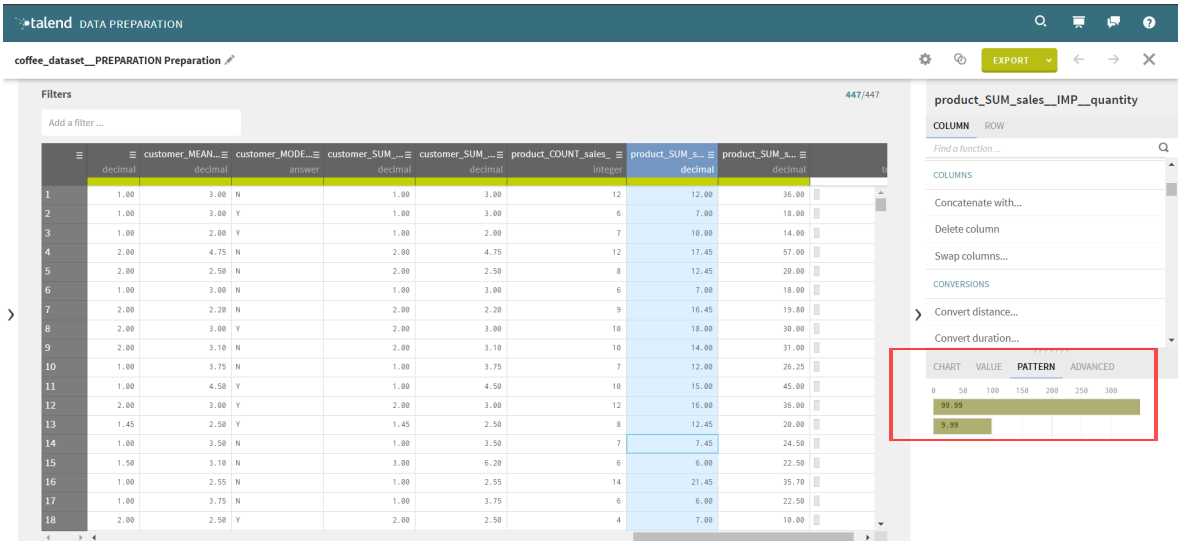


Diagram 2.3.25: Results for rounding decimal points for 'product_SUM_sales__IMP__quantity'

2.4 Model

2.4.1. Unbalanced Class

If the training data has unbalanced classes for the target variable, the model might become biased towards predicting the majority class, performing well on the training set but poorly on a more balanced testing set.

Our training data has unbalanced classes for the target variable (generation and product_group), thus proactive steps were taken to balance the classes before model training in order to address the imbalanced nature of the target variable.

To overcome this issue, we have implemented resampling techniques specifically by using stratified sampling which focuses on generation and product_group as the target variables respectively. As a result, the training dataset included a more equal representation of both classes.

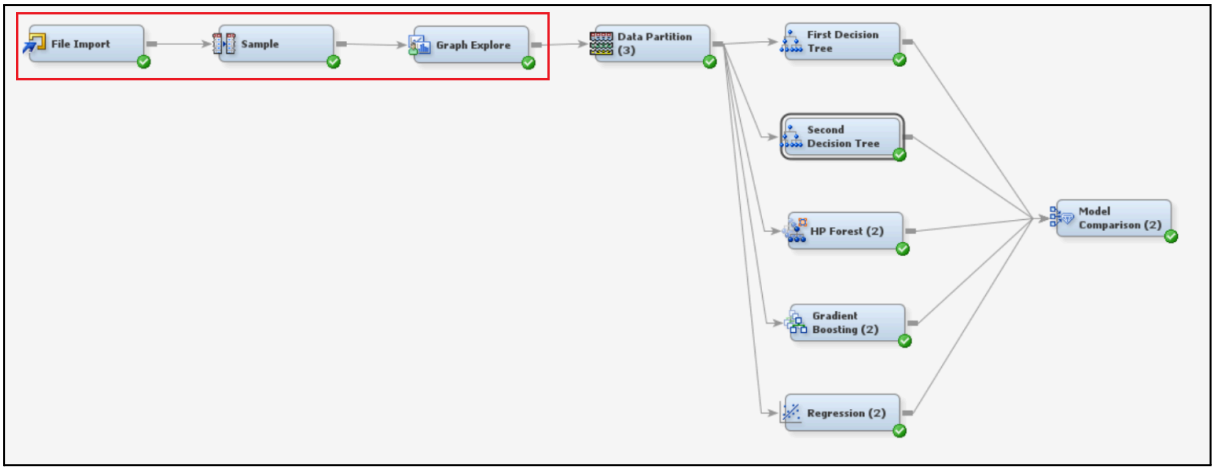


Diagram 2.4.1: Resampling nodes

Train	
Variables	...
Output Type	Data
Sample Method	Stratify
Random Seed	12345
Size	
Type	Percentage
Observations	.
Percentage	100.0
Alpha	0.01
PValue	0.01
Cluster Method	Random
Stratified	
Criterion	Equal
Ignore Small Strata	No
Minimum Strata Size	5

Diagram 2.4.2: Setting to stratify sampling

Result:

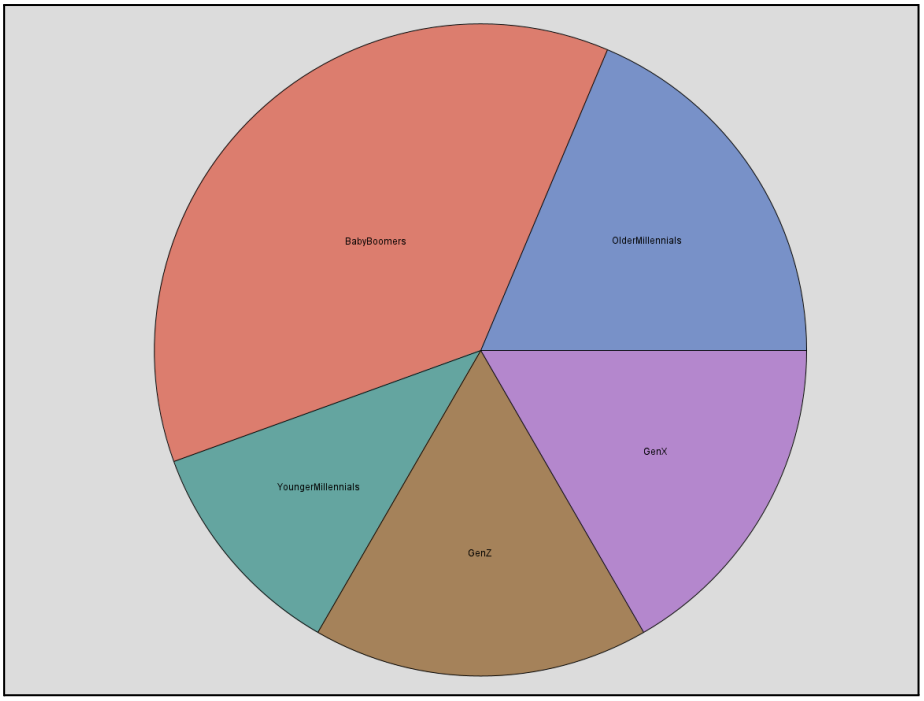
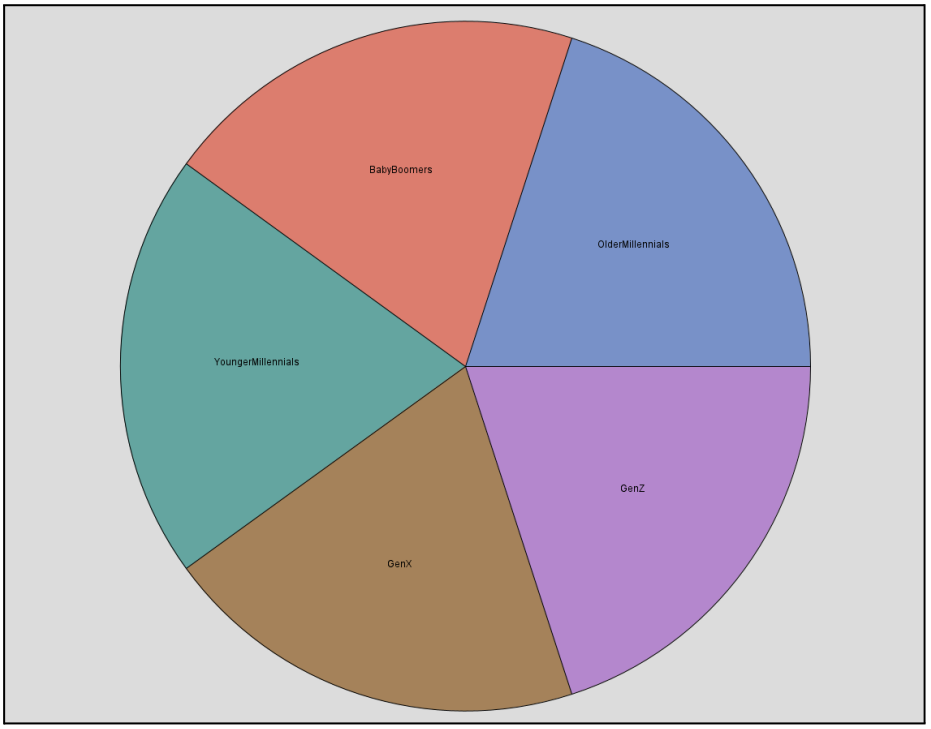
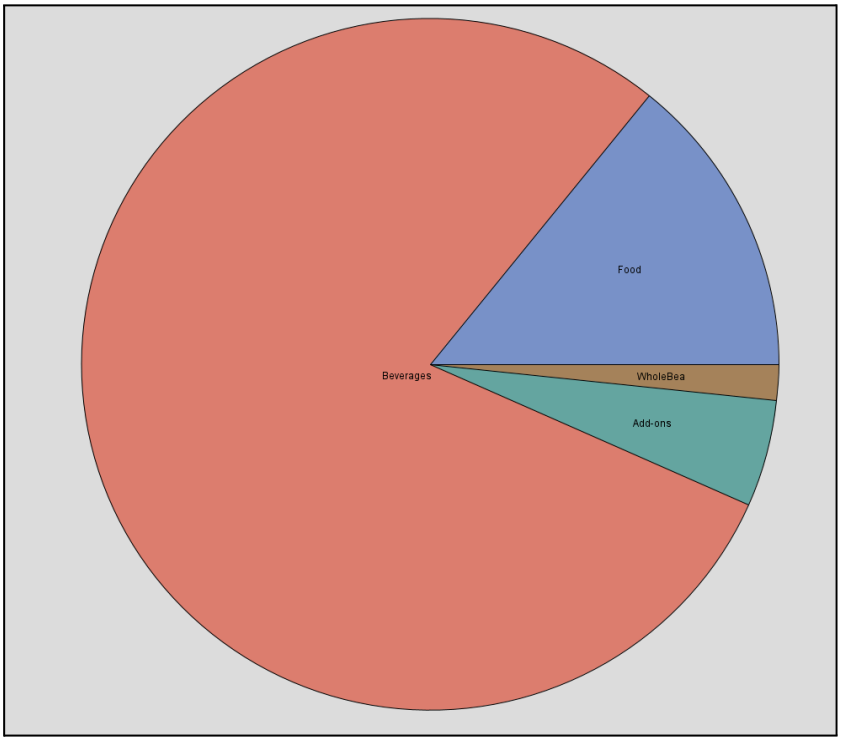
Target Variable: generation
Before

After


Diagram 2.4.3: Results resampling for variable ‘generation’

Target Variable: product_group

Before



After

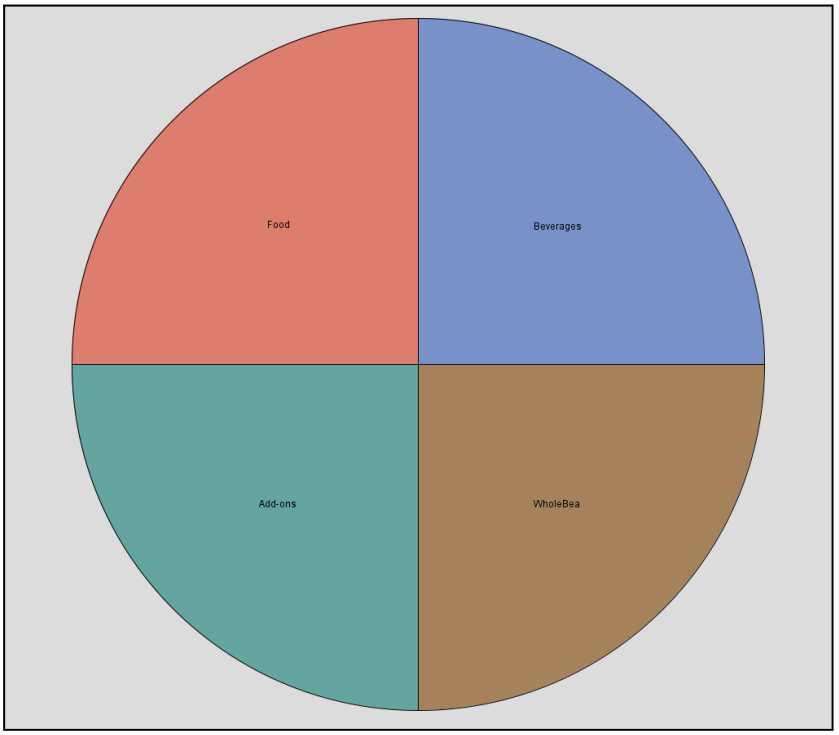


Diagram 2.4.3: Results resampling for variable ‘generation’

2.4.2 Modeling

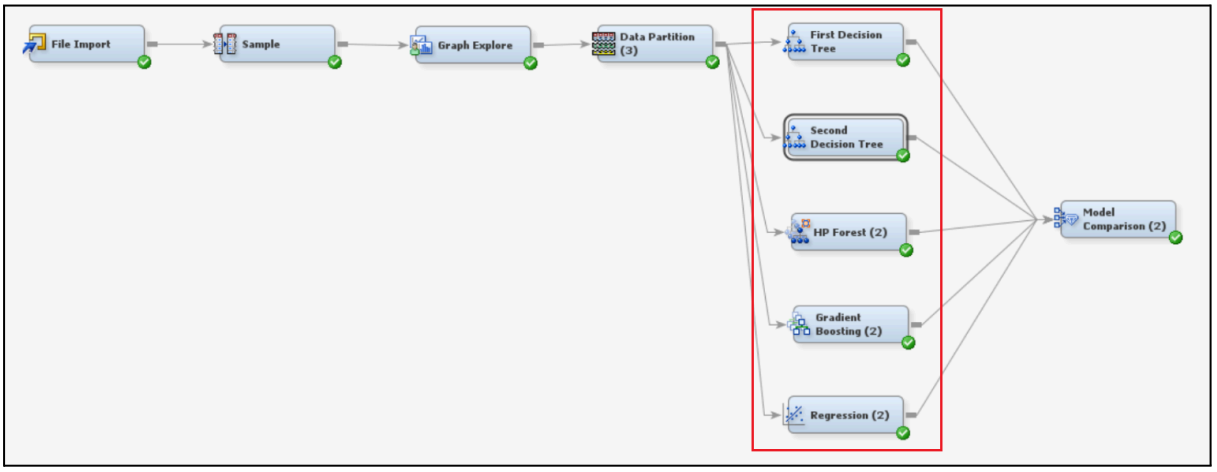


Diagram 2.4.4: Whole Modeling nodes SAS Diagram

Data Set Allocations	
Training	70.0
Validation	30.0
Test	0.0

Diagram 2.4.5: Setting for Data Partition Node

2.4.2.1 Decision Tree

A decision tree is a data segmentation tool, employing simple rules to create hierarchical segments. Rules, based on input values, classify data into segments, yielding interpretability. In this coffee shop project, we classify the target variables of generation and product group. The decision tree's advantage lies in its interpretable node rule.

→ Target variable: generation

For generation aspects, we need to explore product popularity within each generation for optimizing product selection, improving inventory control, and catering to diverse customer preferences across age segments. A Decision Tree aids in achieving this by revealing influential factors guiding product preferences.

By determining variable importance, they identify key features shaping preferences. The segmentation process hierarchically divides data based on generational attributes, revealing nuanced preferences. Decision tree interpretability aids managers in grasping influential factors transparently. Insights from decision trees inform optimization strategies, guiding tailored product selection aligned with generational preferences. This comprehensive approach enhances inventory control and strategic decision-making, ensuring products resonate effectively across diverse generational tastes.

Splitting Rule	
Interval Target Criterion	ProbF
Nominal Target Criterion	ProbChisq
Ordinal Target Criterion	Entropy
Significance Level	0.2
Missing Values	Use in search
Use Input Once	No
Maximum Branch	2
Maximum Depth	6
Minimum Categorical Size	5

Diagram 2.4.6: Setting for First Decision Tree Diagram with target variable 'generation'

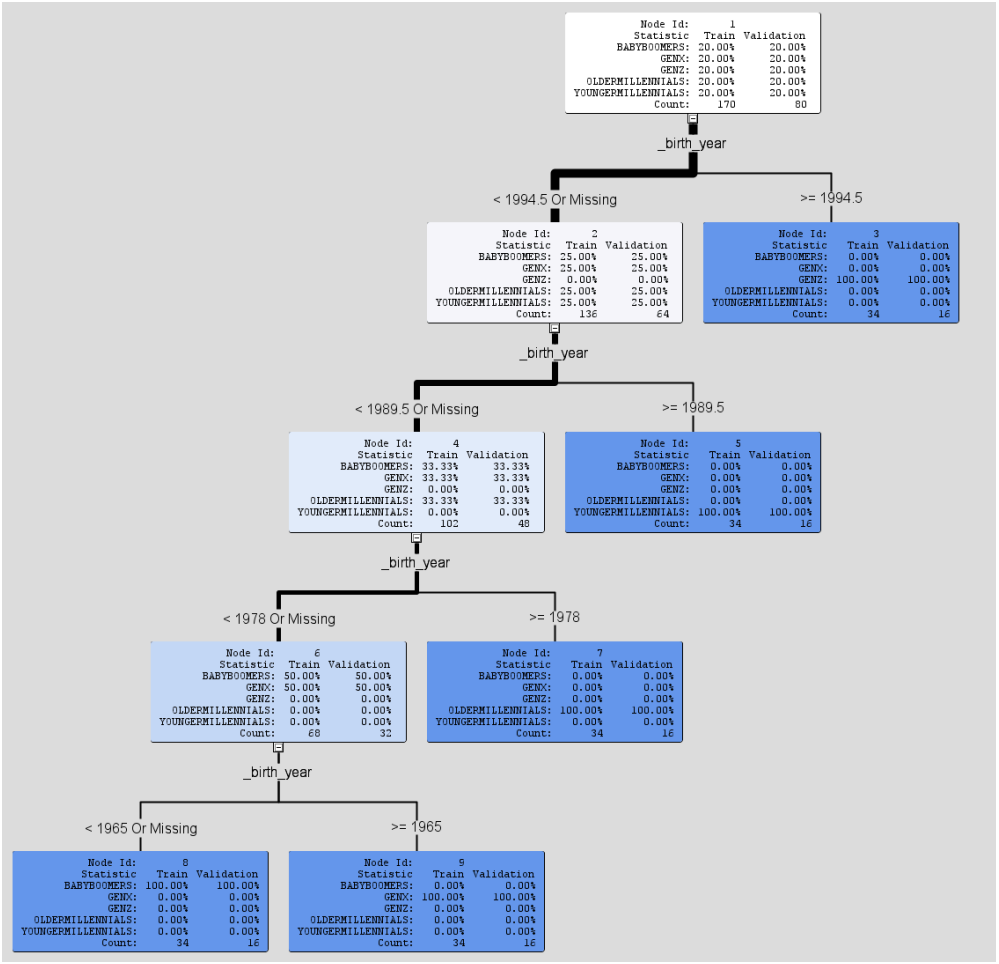


Diagram 2.4.7: Tree Diagram Result of First Decision Tree with target variable 'generation'

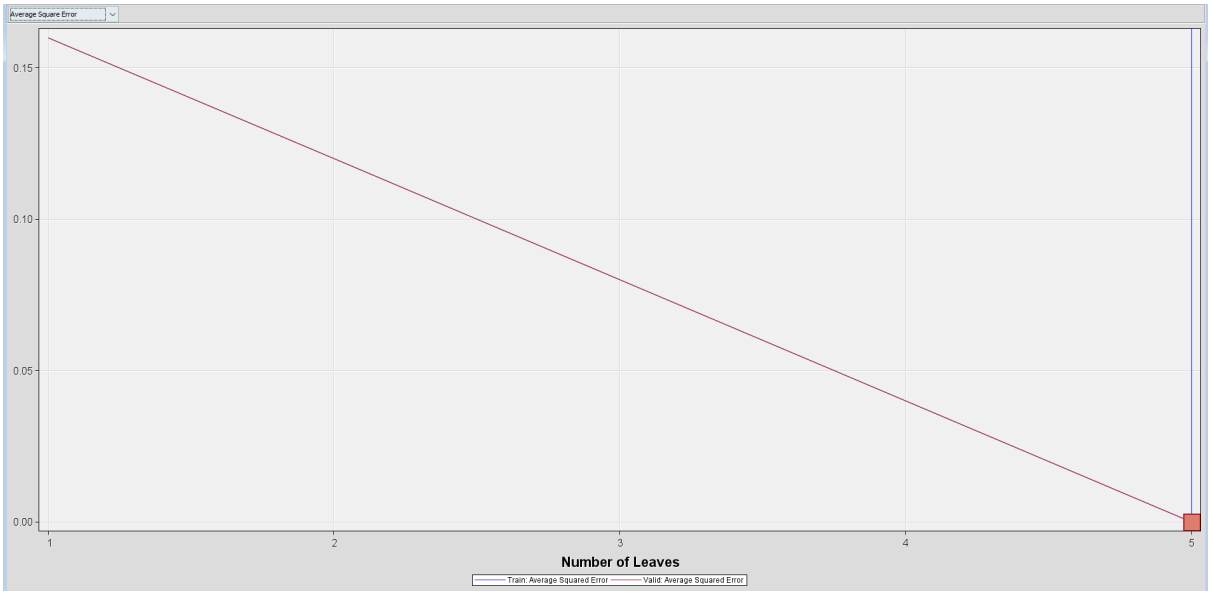
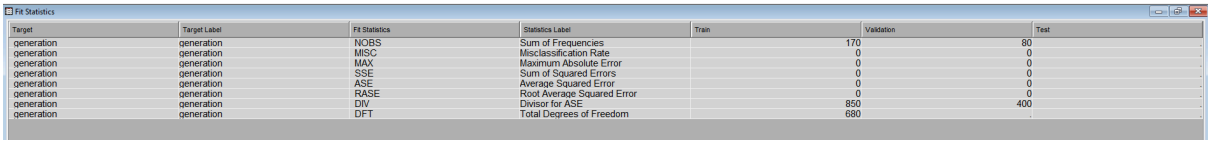


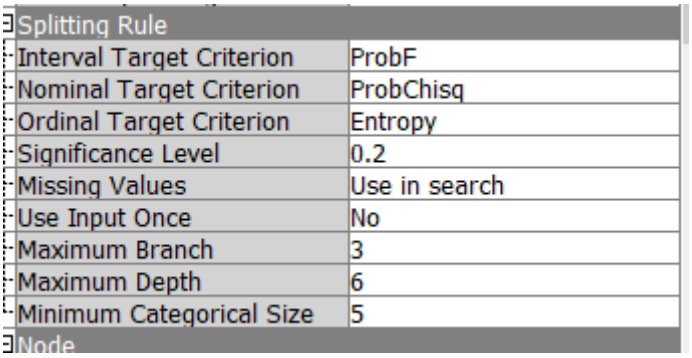
Diagram 2.4.8: Leaf Assessment Result of First Decision Tree with target variable 'generation'



Fit Statistics

Target	Target Label	Fit Statistics	Statistics Label	Train	Validation	Test
generation	generation	NOBS	Sum of Frequencies	170	80	.
generation	generation	MISC	Misclassification Rate	0	0	.
generation	generation	MAX	Maximum Absolute Error	0	0	.
generation	generation	SSE	Sum of Squared Errors	0	0	.
generation	generation	ASE	Average Squared Error	0	0	.
generation	generation	RASE	Root Average Squared Error	0	0	.
generation	generation	DIV	Divisor for ASE	850	400	.
generation	generation	DFT	Total Degrees of Freedom	680	.	.

Diagram 2.4.9: Fit Statistics Result of First Decision Tree with target variable 'generation'



Splitting Rule	
Interval Target Criterion	ProbF
Nominal Target Criterion	ProbChisq
Ordinal Target Criterion	Entropy
Significance Level	0.2
Missing Values	Use in search
Use Input Once	No
Maximum Branch	3
Maximum Depth	6
Minimum Categorical Size	5
Node	

Diagram 2.4.10: Setting for Second Decision Tree Diagram with target variable 'generation'

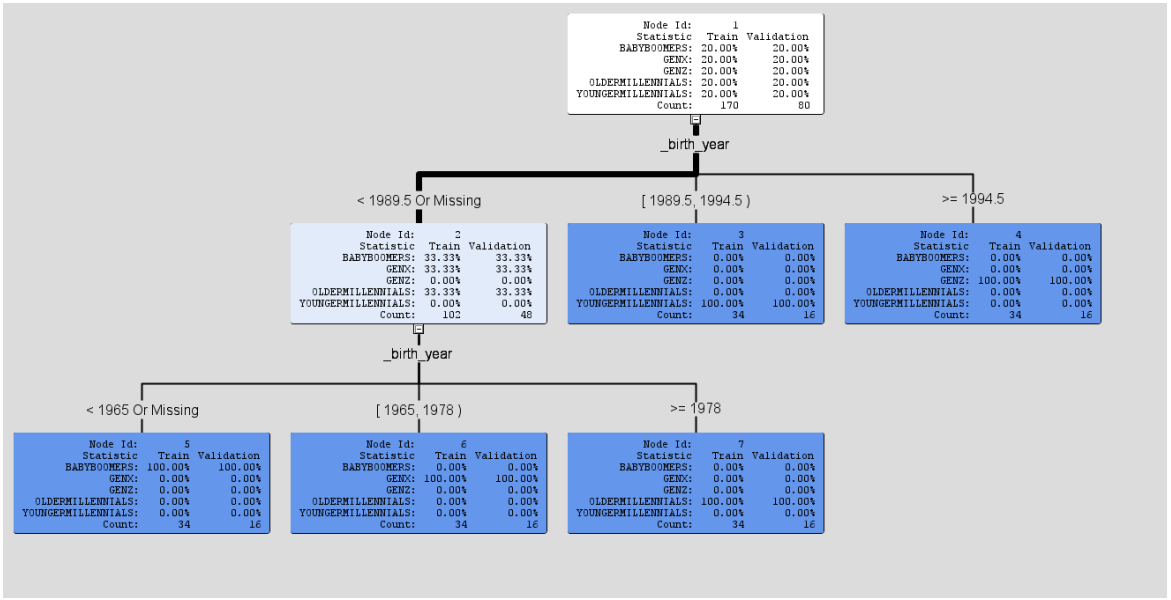


Diagram 2.4.11: Tree Diagram Result of Second Decision Tree with target variable 'generation'

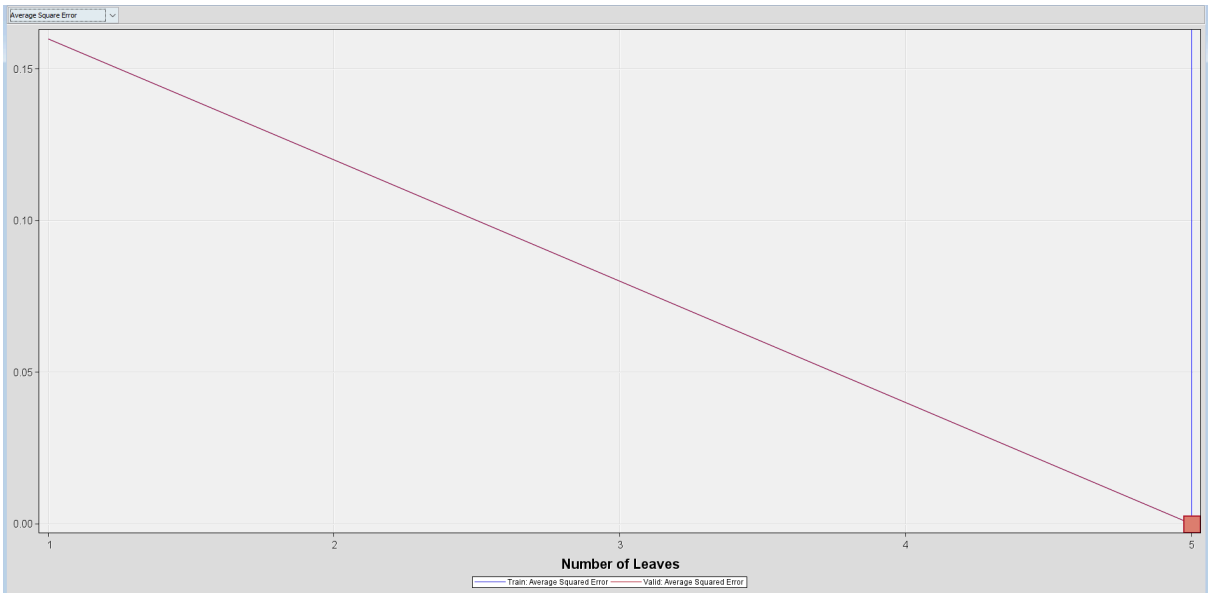
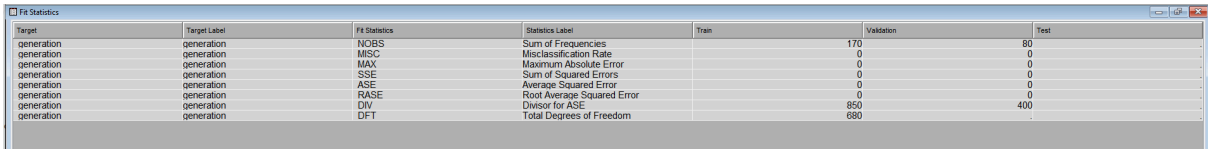


Diagram 2.4.12: Leaf Assessment Result of Second Decision Tree with target variable 'generation'


Fit Statistics

Target	Target Label	Fit Statistics	Statistics Label	Train	Validation	Test
generation	generation	NOBS	Sum of Frequencies	170	80	.
generation	generation	MISC	Misclassification Rate	0	0	.
generation	generation	MAX	Maximum Absolute Error	0	0	.
generation	generation	SSE	Sum of Squared Errors	0	0	.
generation	generation	ASE	Average Squared Error	0	0	.
generation	generation	RASE	Root Average Squared Error	0	0	.
generation	generation	DIV	Divisor for ASE	850	400	.
generation	generation	DFT	Total Degrees of Freedom	680	.	.

Diagram 2.4.13: Fit Statistics Result of Second Decision Tree with target variable 'generation'

→ **Target variable: product_group**

In order to help with resource allocation and strategic promotion synchronization at moments of high demand, it may identify daily trends of coffee purchase, which contributes to improved operational efficiency. In addition, a decision tree can determine the popularity of a product within each generation, facilitating the best possible product selection, inventory management, and catering to a wide range of tastes across age groups in a more focused and effective way.

Splitting Rule	
Interval Target Criterion	ProbF
Nominal Target Criterion	ProbChisq
Ordinal Target Criterion	Entropy
Significance Level	0.2
Missing Values	Use in search
Use Input Once	No
Maximum Branch	2
Maximum Depth	4
Minimum Categorical Size	5

Diagram 2.4.14: Setting for First Decision Tree Diagram with target variable 'product_group'

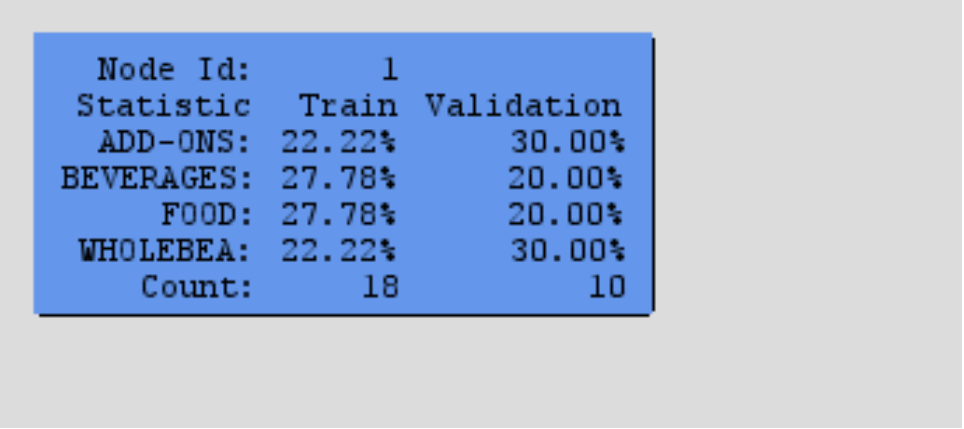


Diagram 2.4.15: Tree Diagram Result of First Decision Tree with target variable 'product_group'

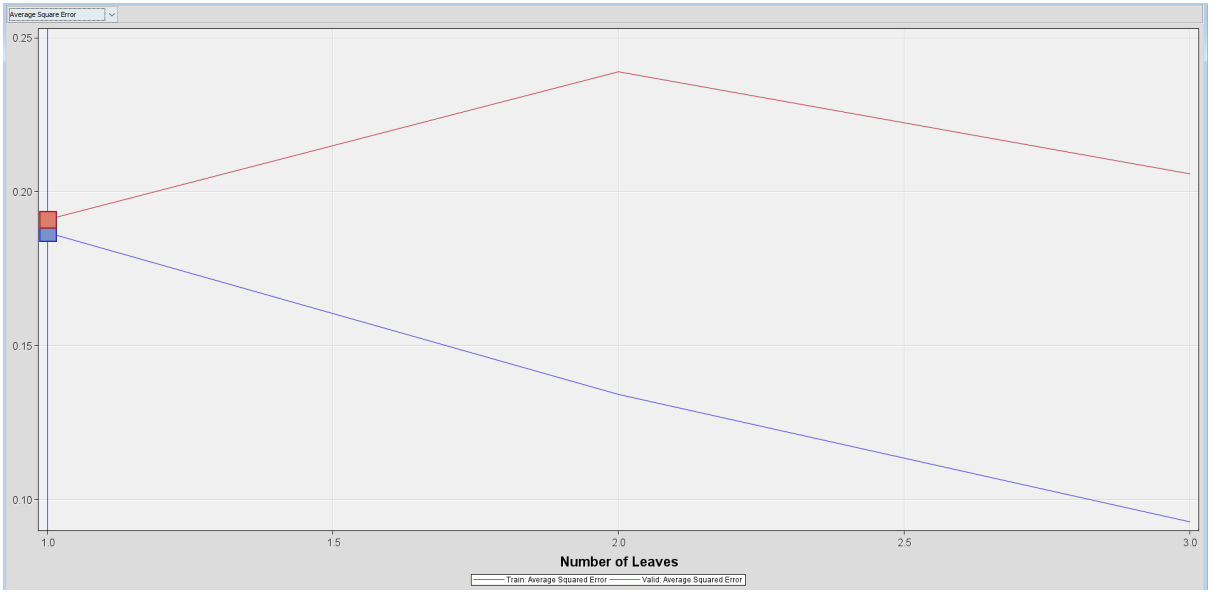
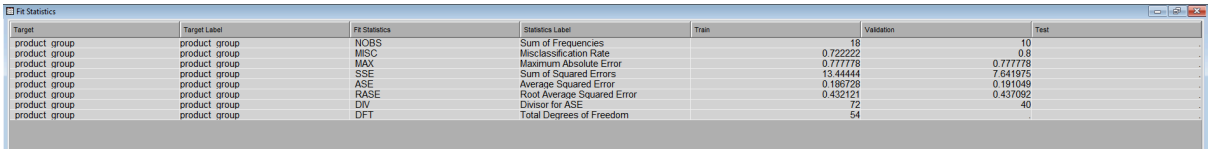


Diagram 2.4.16: Leaf Assessment Result of First Decision Tree with target variable 'product_group'



Target	Target Label	Fit Statistics	Statistics Label	Train	Validation	Test
product_group	product_group	NOBS	Sum of Frequencies	18	10	
product_group	product_group	MISC	Misclassification Rate	0.722222	0.8	
product_group	product_group	MAX	Maximum Absolute Error	0.777778	0.777778	
product_group	product_group	SSE	Sum of Squared Errors	13.44444	7.641975	
product_group	product_group	ASE	Average Squared Error	0.186728	0.191049	
product_group	product_group	RASE	Root Average Squared Error	0.432121	0.437092	
product_group	product_group	DIV	Divisor for ASE	72	40	
product_group	product_group	DFT	Total Degrees of Freedom	54		

Diagram 2.4.17: Fit Statistics Result of First Decision Tree with target variable 'product_group'

Splitting Rule	
Interval Target Criterion	ProbF
Nominal Target Criterion	ProbChisq
Ordinal Target Criterion	Entropy
Significance Level	0.2
Missing Values	Use in search
Use Input Once	No
Maximum Branch	3
Maximum Depth	6
Minimum Categorical Size	5

Diagram 2.4.18: Setting for Second Decision Tree Diagram with target variable 'product_group'

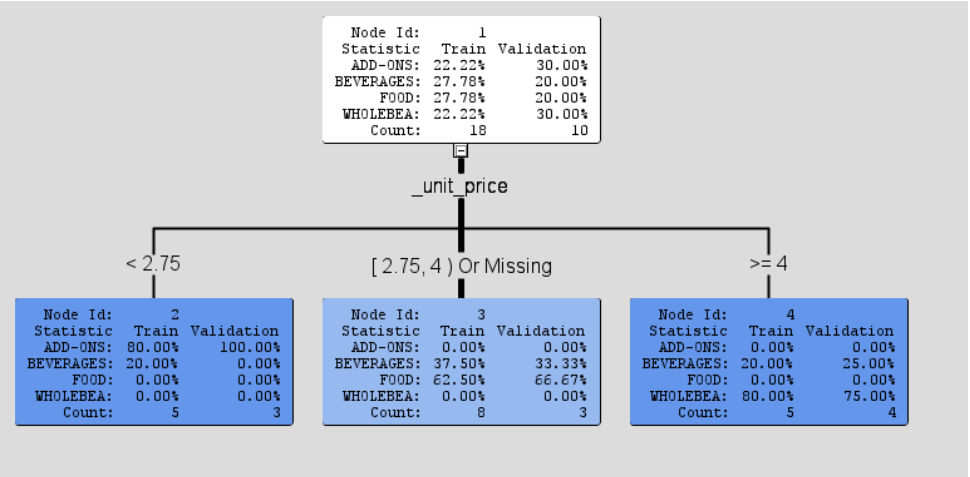


Diagram 2.4.19: Tree Diagram Result of Second Decision Tree with target variable 'product_group'

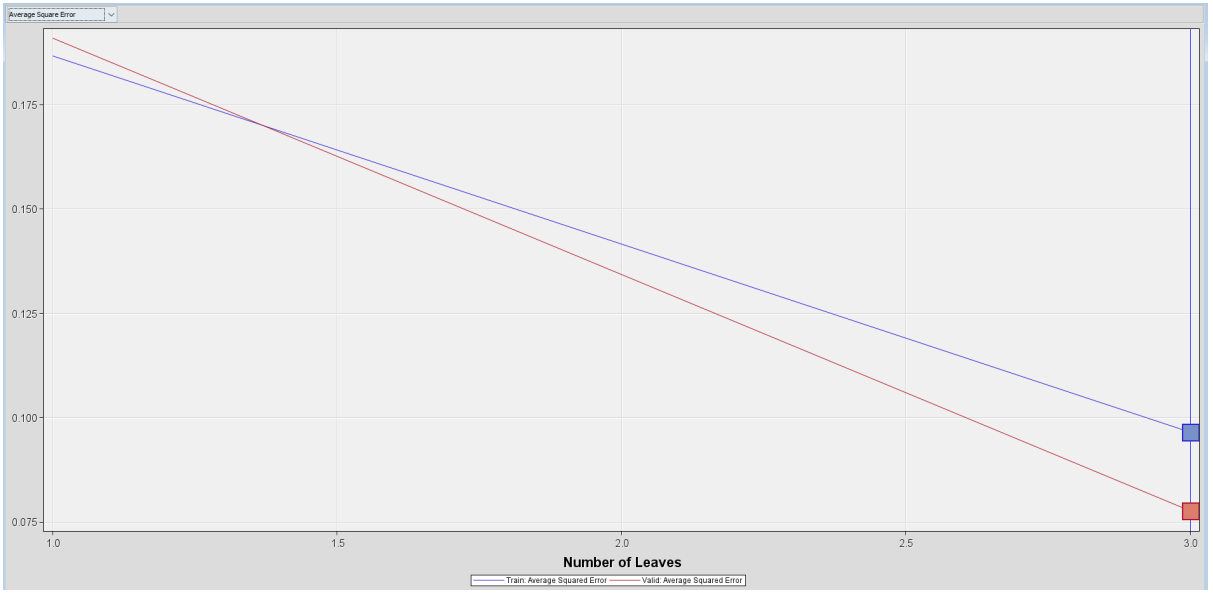
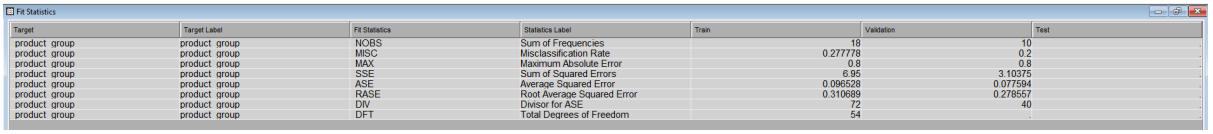


Diagram 2.4.20: Leaf Assessment Result of Second Decision Tree with target variable 'product_group'


Fit Statistics

Target	Target Label	Fit Statistics	Statistics Label	Train	Validation	Test
product_group	product_group	NOBS	Sum of Frequencies	18	10	.
product_group	product_group	MISC	Misclassification Rate	0.277778	0.2	.
product_group	product_group	MAX	Maximum Absolute Error	0.8	0.8	.
product_group	product_group	SSE	Sum of Squared Errors	6.95	3.10375	.
product_group	product_group	ASE	Average Squared Error	0.096528	0.077594	.
product_group	product_group	RASE	Root Average Squared Error	0.310689	0.278557	.
product_group	product_group	DIV	Divisor for ASE	72	40	.
product_group	product_group	DFT	Total Degrees of Freedom	54	.	.

Diagram 2.4.21: Fit Statistics Result of Second Decision Tree with target variable 'product_group'

Decision Tree Analysis

In this SAS Enterprise Miner analysis, we attempted to develop decision tree models to predict two different target variables: "Generation" and "Product Group." For the prediction of "Generation," two decision tree models with different maximum branches were built. Decision Tree 1 revealed a more sophisticated structure with 5 layers that captured subtle patterns within the data, but Decision Tree 2 revealed a shallower depth of 3 levels that emphasized simplicity. This gap in tree depths suggests a trade-off between complexity and interpretability, with a deeper tree risking overfitting but potentially revealing complex relationships. Both decision trees achieve the same performance which is 0 in Average Squared Error for both ‘Train’ and ‘Validation’ sets.

When it came to predicting "Product Group," the models showed varied depths and branches. Decision Tree 1 yielded a single-level tree, indicating a simple decision boundary, but Decision Tree 2 yielded a two-level tree, providing for a little more nuanced representation of the data. The maximum depth chosen in each scenario reflects model interpretability and the required level of granularity in collecting predictive patterns. In result, looking at the Average Squared Error, for Decision Tree 1, scores 0.18 for training and 0.19 for validation, while for Decision Tree 2, scores 0.10 for training and 0.08 for validation.

The differences in tree depths between models require careful analysis of the trade-offs involved. Deeper trees may provide more deep knowledge of the data but may be vulnerable to overfitting, whereas shallower trees may sacrifice complexity for increased interpretability. Besides, from the leaf assessment chart, it found out that the optimum number of leaves are at leaves equal to 5.

2.4.2.2 Random Forest

Random Forest, an ensemble learning method, aggregates predictions from multiple decision trees. Its general function includes improving accuracy, identifying variable importance, handling non-linearity, reducing overfitting, and providing robust insights, making it effective for diverse data analysis tasks.

→ **Target variable: generation**

In the coffee shop project analyzing product popularity across generations, Random Forest, an ensemble learning method, offers enhanced accuracy by aggregating predictions from multiple decision trees. Its ability to identify variable importance aids in understanding influential factors. Moreover, Random Forest excels in capturing non-linear relationships, reducing overfitting, and handling missing values, ensuring robust insights into complex patterns of product popularity and facilitating generalization for strategic decision-making in the dynamic coffee shop environment.

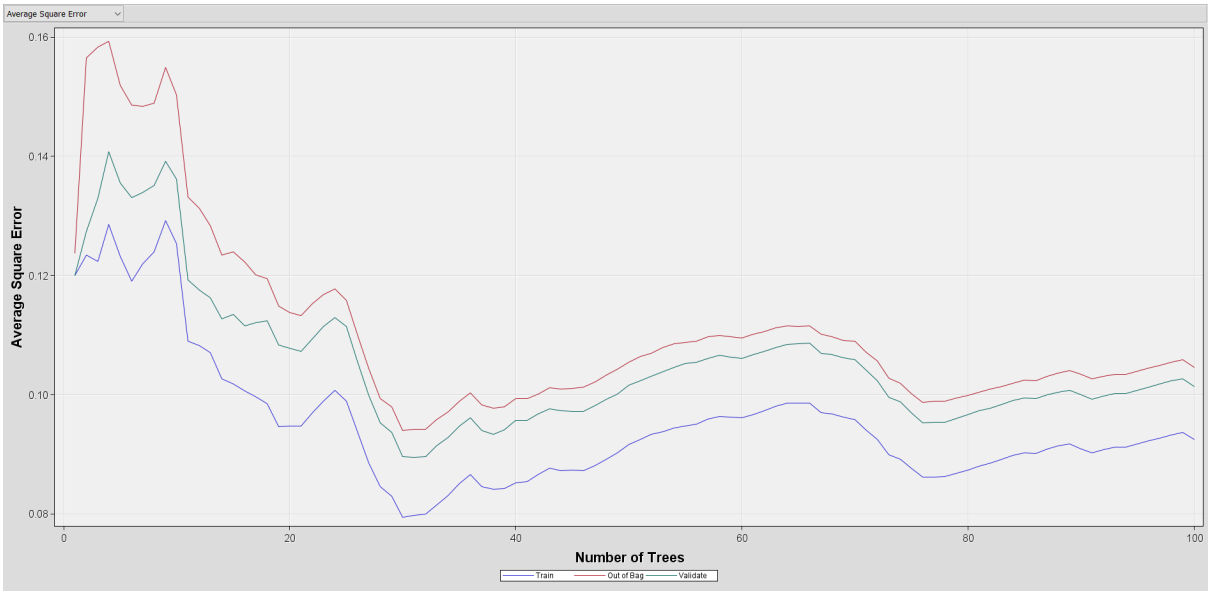


Diagram 2.4.22: Result of Random Forest with target variable 'generation'

Refer to diagram above, the model has significant improvement in performing the validation set and needs more complex technique to capture patterns. But, beyond the 20 trees, the performance declines and may have risk of overfitting.

→ **Target variable: product_group**

Random Forest is good at identifying complex daily patterns of coffee consumption, allocating resources as efficiently as possible, and intelligently timing promotions for peak demand. This improves operational effectiveness by offering analytical insights. Random Forest's ensemble method makes it easier to gain a thorough grasp of product popularity within each generation. This understanding helps to guide the best possible product selection, improved inventory control, and custom tactics for a wide range of client preferences across age groups.

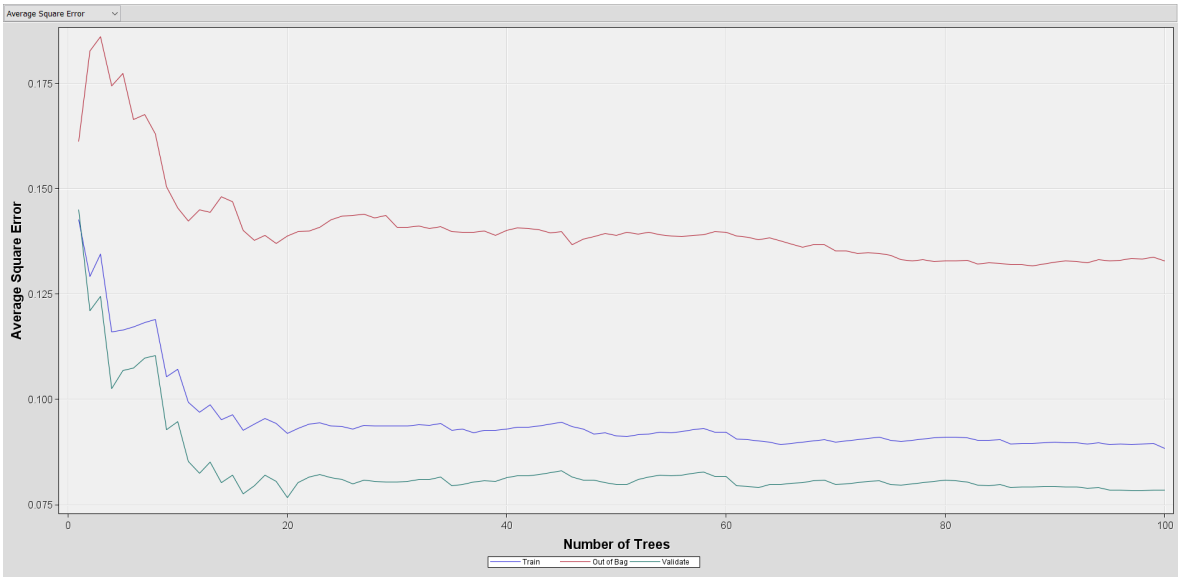


Diagram 2.4.23: Result of Random Forest with target variable 'product_group'

For Random Forest on product_group, the starting higher in error suggests that the model is too simple to show the data complexities. But as the number of trees starting 20, it shows stability which reaches optimal complexity. While it keeps stability, the graph starts capturing noise and data cannot generalize well as the error, train and validation are kept maintained.

Random Forest Analysis

In the analysis of the random forest models for "Generation" as target variable, the chart of average square error vs tree number indicates a significant pattern. The minimum average square error occurs at approximately 30 trees, indicating that this is the point at which the model achieves its optimal balance between accuracy and computational efficiency. Similarly, for "Product Group," the minimum average square error is observed at around 20 trees. For both cases, beyond the minimum point, there is a distinct fluctuation in the error, with both increasing and decreasing patterns observed. This implies that, while an ensemble of trees enhances the model up to a point, adding more trees beyond the optimum does not consistently improve predictive performance. The increasing pattern implies a potential risk of overfitting, where the model might capture noise in the data rather than genuine patterns.

2.4.2.3 Gradient Boosting

Gradient boosting is a machine learning technique that involves three main components: a loss function to be optimized, a weak learner to make predictions, and an additive model to add weak learners to minimize the loss function. The process starts with a base model, often a simple decision tree. The algorithm then iteratively trains new models to correct the errors of the previous models. The final model is the sum of all the models. Building predictive models using gradient boosting is an effective method to do the model.

→ Target variable: generation

In the coffee shop project analyzing product popularity across generations, Gradient Boosting enhances accuracy by sequentially refining decision trees, correcting errors, and improving predictions. Its iterative learning process captures refined patterns in generational preferences, making it invaluable for precise insights. Additionally, it mitigates overfitting and handles complex relationships, contributing to a robust model for strategic decision-making in optimizing product selection and inventory control.

Result

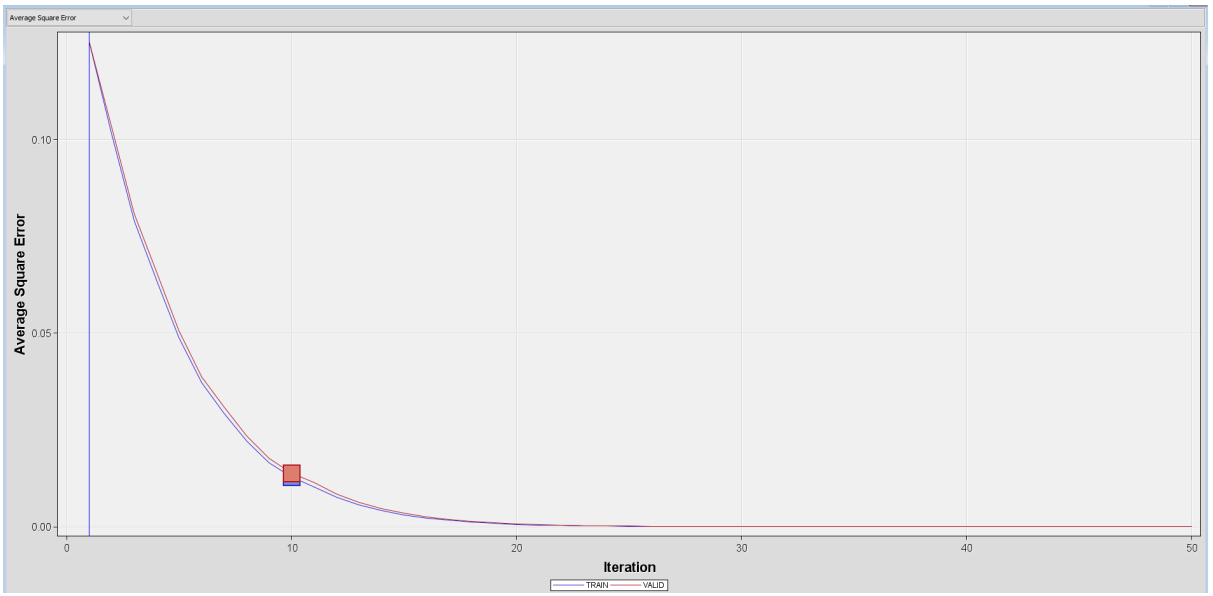


Diagram 2.4.24. Result of Gradient Boosting with target variable 'generation'.

With the result above, the train and valid are at the same level of ASE towards the iteration, showing it not generalizing well for new data. But the decrease after 10 iterations shows that model is performing well on the validation set.

→ Target variable: product_group

Gradient Boosting is efficient at identifying complex daily patterns of coffee consumption, allocating resources as efficiently as possible, and timing promotions for when demand is highest. Gradient Boosting iterative learning process identifies small variations in product popularity within each generation, which informs the best choices, improves

inventory management, and adjusts methods to suit a wide range of client preferences across age groups.

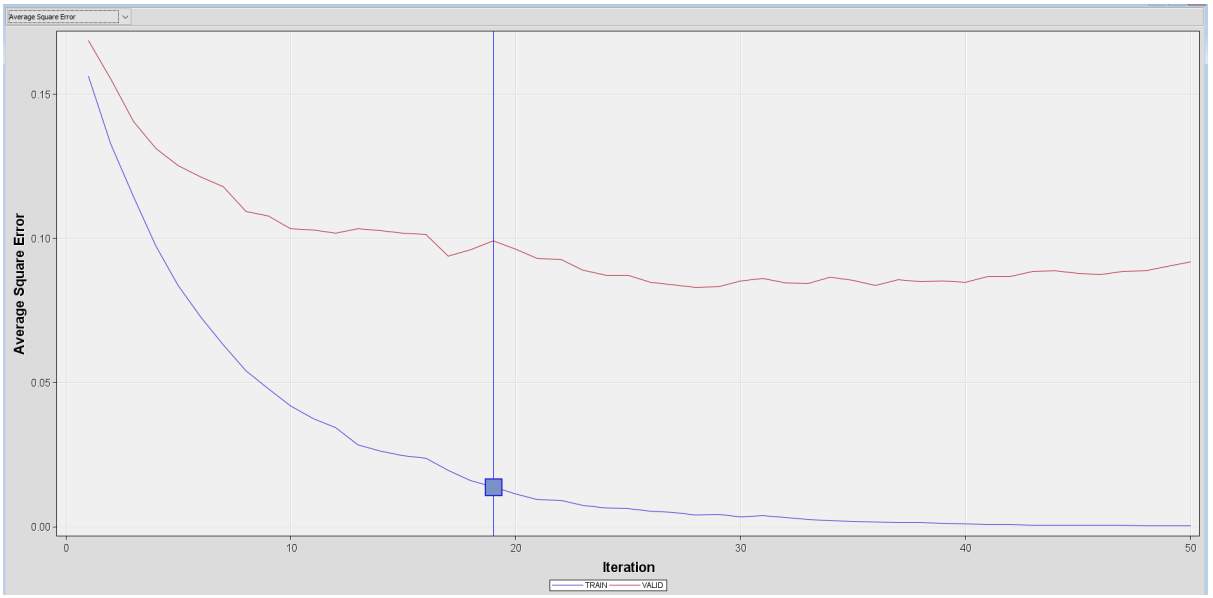


Diagram 2.4.25. Result of Gradient Boosting with target variable product_group.

In the result above, the huge drop on the train set displays the model fitting the training data too close, but for a valid test, it shows the model starts to generalize well to new data. Adding more iteration after 40 does not help improve the model performance.

Gradient Boosting Analysis

In the analysis of gradient boosting models for predicting "Generation," the line chart representing the average square error against the number of iterations reveals various characteristics. The smallest average square error occurs after around 25 iterations, indicating that the model has reached its peak predicted accuracy. However, the following straight-line pattern indicates that additional iterations do not result in significant increases in model performance beyond this point. The result is consistent with the law of diminishing returns, underlining the significance of balancing computing efficiency and prediction accuracy. The straight-line pattern indicates that the model has stabilized and that further iterations are unlikely to improve its capabilities. This understanding is critical for establishing the best configuration for the "Generation" prediction model.

Conversely, in the case of predicting "Product Group," the line chart displays a more detailed pattern. At around 35 iterations, the smallest average square error is seen, representing the point of optimal model performance. The subsequent increasing and decreasing pattern following the minimum point, on the other hand, reveals a more subtle relationship between the number of iterations and average square error. This variation could be attributed to the model navigating a complex space in which extra boosting rounds intermittently contribute to improved predictions while also introducing noise. This result emphasizes the significance of carefully balancing the number of iterations used to capture meaningful patterns while avoiding overfitting.

2.4.2.4 Logistic Regression

Logistic Regression is a statistical method used for binary classification tasks. It models the probability of an event occurring by fitting a logistic curve to the data. It's widely employed in predicting outcomes with two possible classes, providing insights into the relationship between independent variables and the likelihood of a specific outcome.

→ Target variable: generation

By simulating the likelihood of particular behaviors over generations, it helps analyze consumer behavior and makes targeted marketing and product tactics more easily customized. Furthermore, logistic regression works well for grouping consumers into generational groups according to patterns of behavior, giving insights into the age ranges that are most common among the target audience. It can also predict a product's popularity within each generation, which helps to optimize inventory control and product selection based on a range of client preferences. The average change in the response variable's log odds for every unit increase in the predictor variable is represented by the coefficients in a logistic regression model.

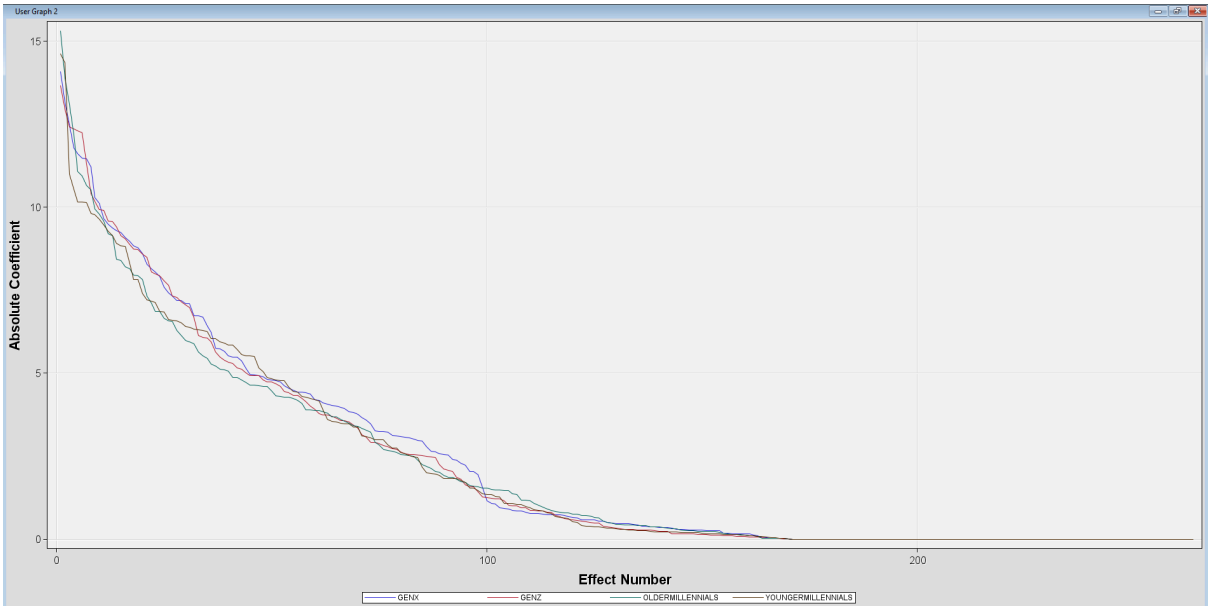


Diagram 2.4.26. Result of Logistic Regression with the target variable 'generation'.

From the result above, Older Millennials show the steepest slope that show the strongest effect on its variable, while other giver flatter slope on weak effect.

→ Target variable: product_group

Logistic regression analysis prior sales data and discovering factors that affect coffee demand to estimate daily coffee purchases. It may also be used to analyze prior sales data and find characteristics that impact consumer preferences across age groups to determine

product popularity by generation. Logistic regression helps coffee businesses improve product selection, inventory control, and consumer tastes across age groups.

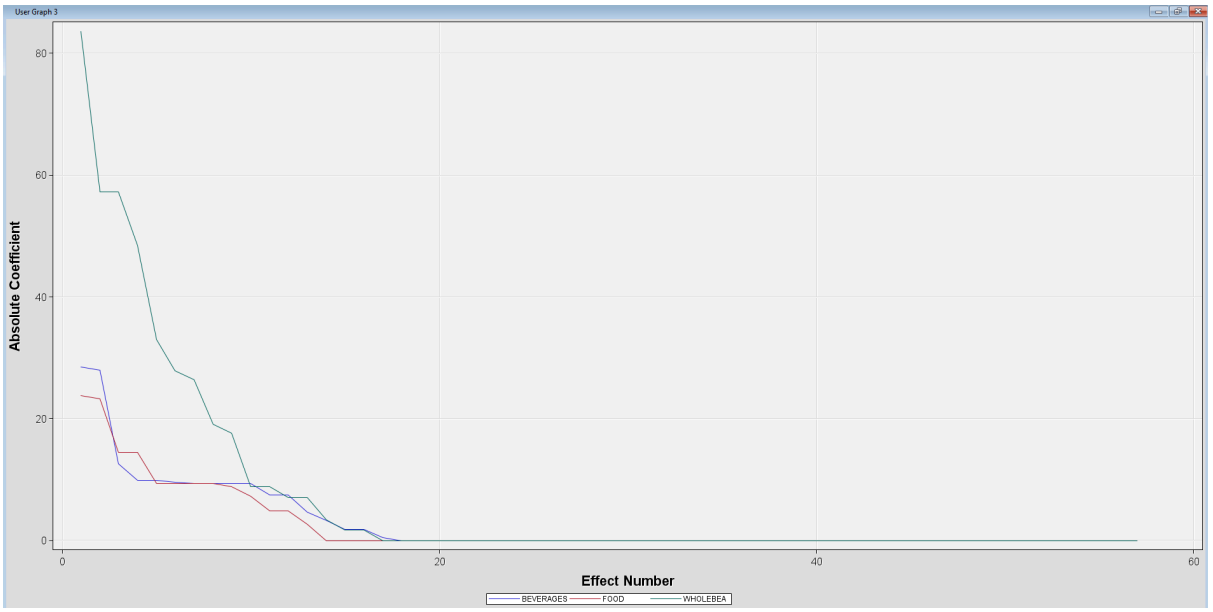


Diagram 2.4.27. Result of Logistic Regression with the target variable 'product_group'.

For the product_group, Whole Bean shows the steepest slope that gives the strongest effect on its coefficient, while Beverages and Food has the lower absolute coefficient and lower slope towards the effects.

Logistic Regression Analysis

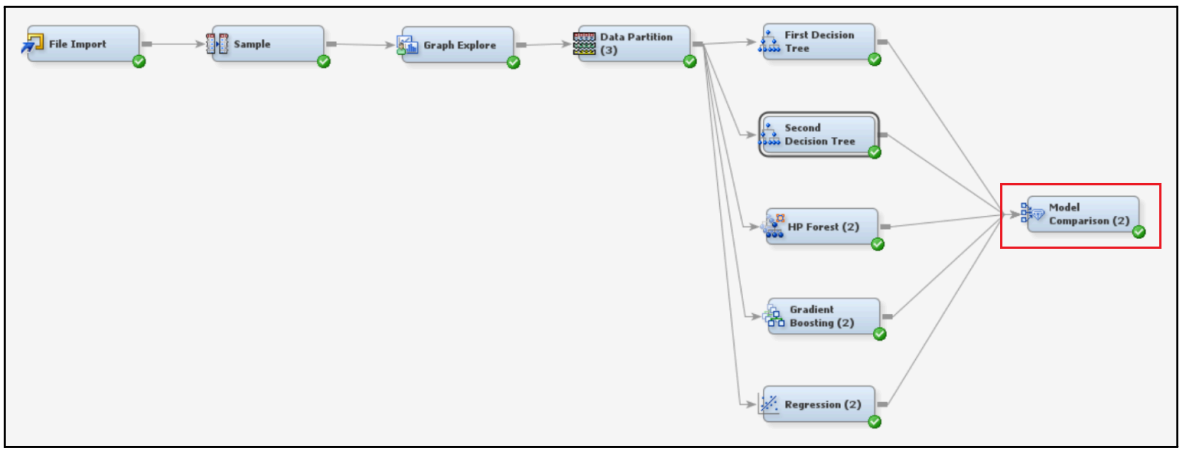
For all the logistic regression results above, the chart displays the line chart of absolute coefficient and effect number. We can differentiate each variable by analyzing the slope line for each. By comparison, as for "Generation", the Older Millennials show the greatest effect as the slope has the highest compared to others. This means we can get insights on the relationship of generation towards the product sales, inventory and and predict product popularity based on the generation, especially in Older Millennials that need to be more focused on, because they contribute more on the shop's product. The other variable may need to get the prediction of the relationship in Gen X, Gen Z and Young Millennials when understanding the product in generation range.

On the other hand, for "Product_Group", by categorizing each product, we can see that the Whole Bean has the highest effect due to the slope being the steepest. The coefficient number from the Whole Bean gets the lowest and maintains an effective number of 19. While the Beverages has the higher coefficient than Food and the effect towards the product is affected by increasing the effect number. From the findings, the company will determine the Whole Bean as the insights to focus on as a more popular product compared

to other products. This can help strategize the planning of its logistics and promotions. For Beverages and Food, as they are also the main products they sell, they can also determine the planning on the promotions and predict the quantity of products to sell.

2.5 Assess

2.5.1 Model Comparison



→ Target variable: generation

Fit Statistics
Model Selection based on Valid: Misclassification Rate (_VMISC_)

Selected Model	Model Node	Model Description	Valid: Misclassification Rate	Train: Average Squared Error	Train: Misclassification Rate	Valid: Average Squared Error
Y	Tree1	Decision Tree (1)	0.0	0.00000	0.000000	0.00000
	Tree2	Decision Tree (2)	0.0	0.00000	0.000000	0.00000
	Boost2	Gradient Boosting (2)	0.0	0.12536	0.000000	0.12536
	HPDMForest2	HP Forest (2)	0.1	0.09249	0.017647	0.10136
	Reg2	Regression (2)	0.8	0.00000	0.000000	0.16000

Figure shows fit statistic for each models

Event Classification Table
Model Selection based on Valid: Misclassification Rate (_VMISC_)

Model Node	Model Description	Data Role	Target	Target Label	False Negative	True Negative	False Positive	True Positive
Tree1	Decision Tree (1)	TRAIN	_generation	generation	0	136	0	34
Tree1	Decision Tree (1)	VALIDATE	_generation	generation	0	64	0	16
Tree2	Decision Tree (2)	TRAIN	_generation	generation	0	136	0	34
Tree2	Decision Tree (2)	VALIDATE	_generation	generation	0	64	0	16
HPDMForest2	HP Forest (2)	TRAIN	_generation	generation	0	136	0	34
HPDMForest2	HP Forest (2)	VALIDATE	_generation	generation	0	64	0	16
Boost2	Gradient Boosting (2)	TRAIN	_generation	generation	0	136	0	34
Boost2	Gradient Boosting (2)	VALIDATE	_generation	generation	0	64	0	16
Reg2	Regression (2)	TRAIN	_generation	generation	0	136	.	34
Reg2	Regression (2)	VALIDATE	_generation	generation	0	.	64	16

Figure on event classification table based on Misclassification Rate

Model Comparison Analysis (Target variable: generation)

In this model comparison analysis, five models were evaluated based on their performance metrics, including misclassification rates and average squared errors, on both the training and validation datasets. The selected models include Decision Tree (1), Decision Tree (2), Gradient Boosting (2), HP Forest (2), and Regression (2). Notably, Decision Tree (1) and Decision Tree (2) exhibited impeccable performance, achieving a flawless 0.0 misclassification rate and an average squared error of 0.00000 on the validation dataset. This suggests that these decision tree models accurately captured the underlying patterns in the data.

Gradient Boosting (2) and HP Forest (2) also demonstrated strong performance, with low misclassification rates of 0.0 and 0.1, respectively, and competitive average squared errors (0.12536 and 0.09249). These models, leveraging ensemble techniques, showcased their ability to generalize well to unseen data.

On the other hand, Logistic Regression (2) exhibited a higher misclassification rate of 0.8 on the validation set, indicating potential challenges in accurately classifying instances. However, it achieved a relatively lower average squared error of 0.000, showcasing its effectiveness in regression tasks.

Further examination of additional fit statistics revealed comprehensive insights into the discriminatory power, information criteria, and model complexity for each selected model. Noteworthy is the strong performance of all models in terms of training metrics, with perfect misclassification rates and low average squared errors, indicating their proficiency in fitting the training data.

In conclusion, the choice of the most suitable model depends on the specific objectives of the modeling task. Decision Tree (1) and Decision Tree (2) stand out for their impeccable performance, while Gradient Boosting (2) and HP Forest (2) present strong contenders. The higher misclassification rate of Logistic Regression (2) warrants careful consideration, and further exploration into interpretability and practical implications will guide the final model selection. Overall, this detailed model comparison provides a nuanced understanding of the strengths and considerations associated with each model, aiding in the informed decision-making process.

→ **Target variable: product_group**

Different models, such as decision trees, random forests, and gradient boosting, were used for the product_group analysis in order to find trends in product popularity. By emphasizing significant variables, decision trees offered interpretability, and through group learning, random forests and gradient boosting improved prediction accuracy. In order to identify which model best reflected the variations of product popularity across several coffee shop product categories, the models were compared based on accuracy, precision, and recall.

Fit Statistics
Model Selection based on Valid: Misclassification Rate (_VMISC_)

Selected Model	Model Node	Model Description	Valid: Misclassification Rate	Train: Average Squared Error	Train: Misclassification Rate	Valid: Average Squared Error
Y	HPDMForest3	HP Forest (2)	0.1	0.08844	0.16667	0.07844
	Tree2	Decision Tree (2)	0.2	0.09653	0.27778	0.07759
	Boost3	Gradient Boosting (2)	0.2	0.13274	0.05556	0.15530
	Reg3	Regression (2)	0.7	0.00000	0.00000	0.16312
	Tree1	Decision Tree (1)	0.8	0.18673	0.72222	0.19105

Figure shows fit statistic for each models

Event Classification Table
Model Selection based on Valid: Misclassification Rate (_VMISC_)

Model Node	Model Description	Data Role	Target	Target Label	False Negative	True Negative	False Positive	True Positive
Tree1	Decision Tree (1)	TRAIN	_product_group	product_group	4	14	0	0
Tree1	Decision Tree (1)	VALIDATE	_product_group	product_group	3	7	0	0
Tree2	Decision Tree (2)	TRAIN	_product_group	product_group	0	13	1	4
Tree2	Decision Tree (2)	VALIDATE	_product_group	product_group	0	6	1	3
HPDMForest3	HP Forest (2)	TRAIN	_product_group	product_group	0	14	0	4
HPDMForest3	HP Forest (2)	VALIDATE	_product_group	product_group	0	7	0	3
Boost3	Gradient Boosting (2)	TRAIN	_product_group	product_group	0	14	0	4
Boost3	Gradient Boosting (2)	VALIDATE	_product_group	product_group	0	7	0	3
Reg3	Regression (2)	TRAIN	_product_group	product_group	.	14	.	4
Reg3	Regression (2)	VALIDATE	_product_group	product_group	2	6	1	1

Figure on event classification table based on Misclassification Rate

Model Comparison Analysis (Target variable: product_group)

In this comprehensive model comparison analysis, five distinct models were meticulously evaluated based on various fit statistics, with a focus on misclassification rates and average squared errors across both training and validation datasets. The selected models encompass HP Forest, Decision Trees, Gradient Boosting, and Logistic Regression.

HP Forest (2) exhibited noteworthy performance with a validation misclassification rate of 0.1 and a corresponding average squared error of 0.08844. This model demonstrated a robust ability to generalize well to unseen data, as evidenced by its low misclassification rate.

Decision Tree (2) and Gradient Boosting (2) presented competitive results, each with a validation misclassification rate of 0.2 and corresponding average squared errors of 0.09653 and 0.13274, respectively. These models showcased a balanced trade-off between accuracy and model complexity.

Logistic Regression (2), however, demonstrated a higher validation misclassification rate of 0.7, indicating potential challenges in accurately classifying instances. Despite this, it achieved a commendable average squared error of 0.16312, suggesting effectiveness in regression tasks.

In contrast, Decision Tree (1) exhibited the highest validation misclassification rate of 0.8, potentially signaling overfitting to the training data. The associated average squared error was 0.000, indicating a compromise in predictive accuracy.

In terms of training performance, HP Forest (2) stood out with a perfect misclassification rate and low average squared error on the validation dataset. Decision Tree (2) and Gradient Boosting (2) displayed competitive performance, while Logistic Regression (2) and Decision Tree (1) exhibited higher misclassification rates, suggesting potential challenges in training data fitting.

Ultimately, the selection criteria based on the misclassification rate favored HP Forest (2). Decision Tree (2) and Gradient Boosting (2) emerged as balanced alternatives, showcasing good generalization capabilities. The higher misclassification rates observed in Logistic Regression (2) and Decision Tree (1) warrant careful consideration, prompting further exploration into their interpretability and practical implications.

In conclusion, this detailed model comparison provides valuable insights into the strengths and considerations associated with each model, facilitating an informed decision-making process based on the specific goals and requirements of the modeling task.

3.0 Conclusion and Future Works

In conclusion, this project has been effective in addressing important difficulties that are present in the coffee shop business by utilizing modern analytical methodologies and technologies. Through the investigation of consumer behavior across generations, useful insights have been obtained, which have been utilized in the customisation of marketing tactics and product offers. The study of daily coffee purchase patterns has resulted in an increase in operational efficiency, which has led to the efficient distribution of resources and the synchronization of promotions with periods of strong demand.

Additionally, the examination into the popularity of items across various age groups has helped to the optimisation of product selection as well as the improvement of inventory control. The adoption of Talend Data Preparation, SAS Enterprise Miner, and Feature Tools in accordance with the SEMMA methodology in order to handle complicated activities linked with consumer behavior analysis and demand forecasting has proved the adaptability of these tools in handling these tasks.

Going forward, there are opportunities for further improvement and refinement that may be looked out for. The accuracy of demand and consumer behavior projections may increase with the use of more advanced machine learning techniques and algorithms in predictive modeling. Investing in real-time data updates and exploring new data sources can help the system respond more quickly to fluctuations in the market.

Furthermore, a more in-depth investigation into the seasonal purchasing patterns might give deeper understanding into the means by which inventory management can be optimized throughout particular time periods. The flexibility of the model to change customer preferences will be ensured by collaborations with other data sources and ongoing monitoring of changes in the sector.

In conclusion, the groundwork that was set by this project provides a solid basis for continued improvements and developments in coffee shop operations. This project places an emphasis on the continual evolution and adaptation that is necessary to fulfill the ever-changing demands the market places on businesses.

4.0 References

Bhandari, P. (2023, June 21). *Descriptive statistics: Definitions, types, examples*. Scribbr. https://www.scribbr.com/statistics/descriptive-statistics/

Brownlee, J. (2020, August 14). A gentle introduction to the gradient boosting algorithm for machine learning. MachineLearningMastery.com. https://machinelearningmastery.com/gentle-introduction-gradient-boosting-algorithm-machine-learning/

Chang, J. (2019, November 8). *Coffee shop sample data (11.1.3+)*. Kaggle. https://www.kaggle.com/datasets/ylchang/coffee-shop-sample-data-1113/data

Fontanella, C. (2022, August 18). *A beginner's Guide to Customer Behavior Analysis*. HubSpot Blog. https://blog.hubspot.com/service/customer-behavior-analysis

Insider Learning Machines. (2023, November 6). How to interpret decision trees with 1 simple example. Inside Learning Machines. https://insidelearningmachines.com/interpret_decision_trees/

Iqbal, J. (2022, August 16). *What are the best statistical models to use for demand forecasting?*. Causometrix. https://www.causometrix.com/what-are-the-best-statistical-models-to-use-for-demand-forecasting/

Javatpoint. (n.d.). Machine learning random forest algorithm - javatpoint. www.javatpoint.com. https://www.javatpoint.com/machine-learning-random-forest-algorithm

Lawton, G., Burns, E., & Rosencrance, L. (2022, January 20). What is logistic regression? - definition from Searchbusinessanalytics. Business Analytics. https://www.techtarget.com/searchbusinessanalytics/definition/logistic-regression

Mab, MabMab 4111 silver badge33 bronze badges, rolando2rolando2 12.2k11 gold badge4242 silver badges6161 bronze badges, RickyBRickyB 1, & AshOfFireAshOfFire 57033 silver badges1010 bronze badges. (1964, May 1). How to visualise coefficients of a binomial logistic regression?. Cross Validated. https://stats.stackexchange.com/questions/342627/how-to-visualise-coefficients-of-a-binomial-logistic-regression

Saini, A. (2024, January 5). Decision tree - a step-by-step guide. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/08/decision-tree-algorithm/

5.0 Appendix

Github Link

https://github.com/zrlxkai/DMGroupProject

YouTube Link

https://youtu.be/I_ox30Zt9nw

Proof of our dedication and effort into this assignment ☺: