Coffee Sales Data Mining Analysis

Presented by Group 1

Name	ID
Bhone Myint San	6520225
Lu Phone Maw	6511157
Lut Lat Aung	6511163
Myat Min Phyo	6511125
Yel Lin	6520242

Content

Introduction

Cafe's sale Dataset from Kaggle.

https://www.kaggle.com/datasets/ahmedmohamed2003/cafe-sales-data-for-cleaning-training

kaggle

orange
DATA MINING

How do we collect our data?

Cafe's sale Dataset from Kaggle.

https://www.kaggle.com/datasets/ahmedmohamed2003/cafe-sales-data-for-cleaning-training

We collected Cafe's sale Dataset from Kaggle website. The Dataset includes numerical and categorical

kaggle

orange
DATA MINING

Data Attribute Categorization

Numerical

- Quantity
- Price Per Unit
- Total Spent

Categorical

- Item
- Payment Method
- Location

Text

- Transaction ID

Time

- Transaction Date

Why convert some features into numerical and time?

✅ Quantity, Price Per Unit, Total Spent

N *Quantity (reinterpreted as numeric)*

N *Price Per Unit (reinterpreted as numeric)*

N *Total Spent (reinterpreted as numeric)*

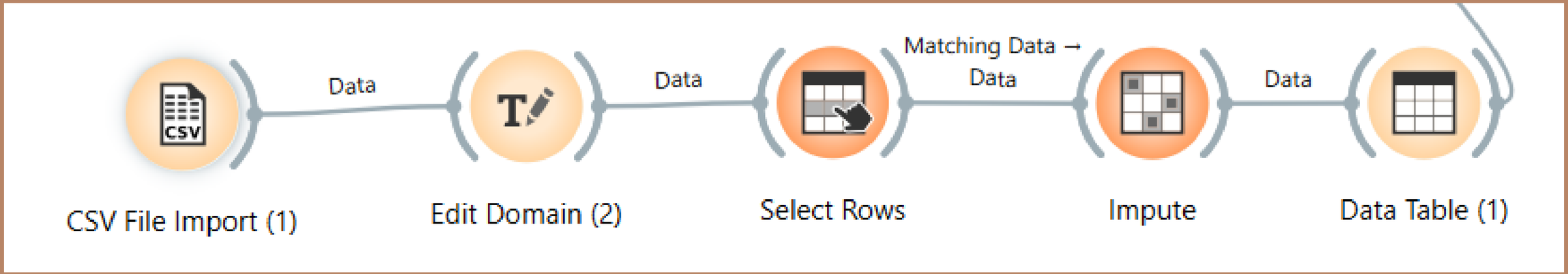
- Helps track sales volume (Quantity), pricing trends (Price Per Unit), and revenue calculations (Total Spent).
- Enables analysis of customer spending patterns, discounts, and profitability of menu items.

✅ Transaction Date

T *Transaction Date (reinterpreted as time)*

- Helps understand seasonal variations in café sales and identify high-traffic times.
- Useful for forecasting demand and managing inventory efficiently.

Data Cleaning



Orange Workflow

- **CSV File - Dirty Data set which include more than 9 thousands instances**
- **We convert some features such as quantity, price per unity and total spent into numerical using edit domain**

Data Cleaning

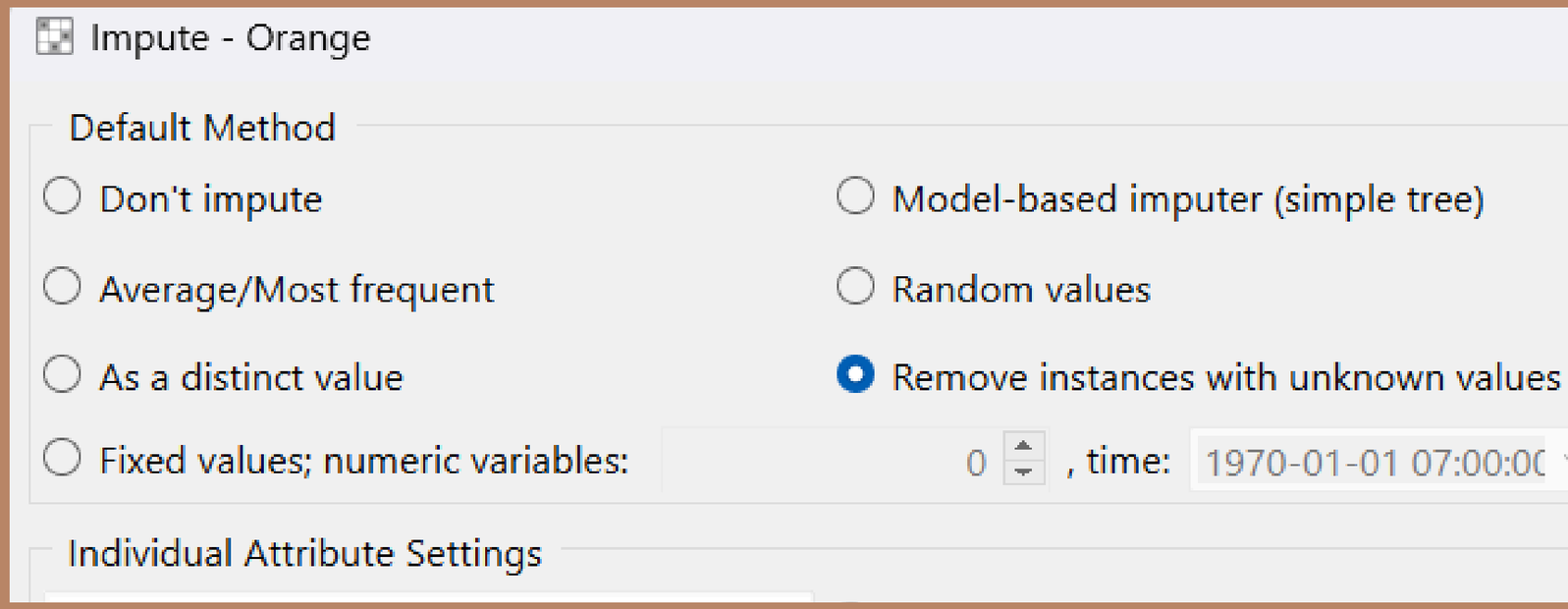
	Transaction ID	ransaction Dat	Item	Quantity	Price Per Unit	Total Spent	ayment Metho	Location
1	TXN_1961373	2023-09-08	Coffee	2	2.0	4.0	Credit Card	Takeaway
2	TXN_4977031	2023-05-16	Cake	4	3.0	12.0	Cash	In-store
3	TXN_4271903	2023-07-19	Cookie	4	1.0	ERROR	Credit Card	In-store
4	TXN_7034554	2023-04-27	Salad	2	5.0	10.0	UNKNOWN	UNKNOWN
5	TXN_3160411	2023-06-11	Coffee	2	2.0	4.0	Digital Wallet	In-store
6	TXN_2602893	2023-03-31	Smoothie	5	4.0	20.0	Credit Card	?
7	TXN_4433211	2023-10-06	UNKNOWN	3	3.0	9.0	ERROR	Takeaway
8	TXN_6699534	2023-10-28	Sandwich	4	4.0	16.0	Cash	UNKNOWN
9	TXN_4717867	2023-07-28	?	5	3.0	15.0	?	Takeaway

Before Cleaning

From observation, we can see that there are Unknown , Error and missing values (?) in both categorical and numerical entries before properly cleaned. There are over 9 thousand data, most of which are unusable and doesn't work well with in our datamining analysis.

Data Cleaning

- **Remove “Unknown” “Error” Values from the dataset using select rows**
- **Make Transaction Date defined to avoid missing date**
- **used the Impute function to deal with missing or incorrect values**



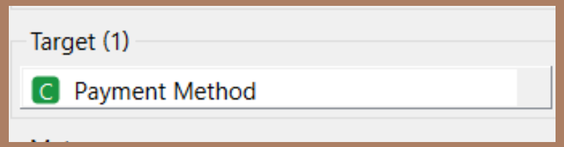
Data Cleaning

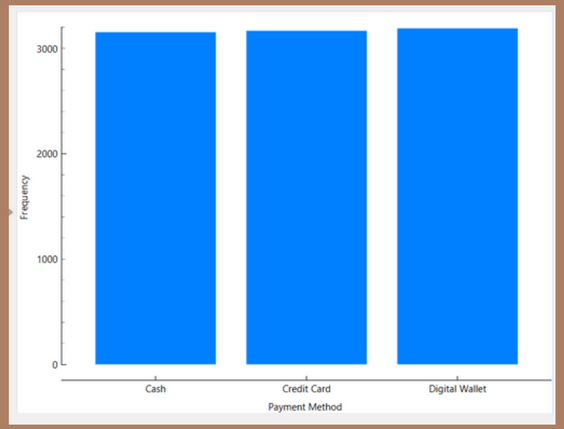
	Transaction ID	ransaction Dat	Item	Quantity	Price Per Unit	Total Spent	ayment Metho	Location
1	TXN_1961373	2023-09-08 0...	Coffee	2	2	4	Credit Card	Takeaway
2	TXN_4977031	2023-05-16 0...	Cake	4	3	12	Cash	In-store
3	TXN_3160411	2023-06-11 0...	Coffee	2	2	4	Digital Wallet	In-store
4	TXN_2548360	2023-11-07 0...	Salad	5	5	25	Cash	Takeaway
5	TXN_7619095	2023-05-03 0...	Sandwich	2	4	8	Cash	In-store
6	TXN_2847255	2023-11-15 0...	Salad	3	5	15	Credit Card	In-store
7	TXN_6769710	2023-02-24 0...	Juice	2	3	6	Cash	In-store
8	TXN_3709394	2023-01-15 0...	Juice	4	3	12	Cash	Takeaway
9	TXN_3567645	2023-03-30 0...	Smoothie	4	4	16	Credit Card	Takeaway

After Cleaning

There are no Unknown, Error and missing values left in the dataset. To work well with our datamining analysis models, we have removed the rows with those values . The cleaned data is around 3 thousands

Classification





- Use “Payment Method” as Target value
- Payment Method has balance Distribution

Model	AUC	CA	F1	Prec	Recall	MCC
SVM	0.508	0.330	0.208	0.222	0.330	-0.005
Neural Network	0.500	0.331	0.330	0.331	0.331	-0.004
Logistic Regression	0.494	0.329	0.327	0.329	0.329	-0.006
Random Forest	0.491	0.327	0.327	0.327	0.327	-0.009
Gradient Boosting	0.490	0.323	0.323	0.323	0.323	-0.016

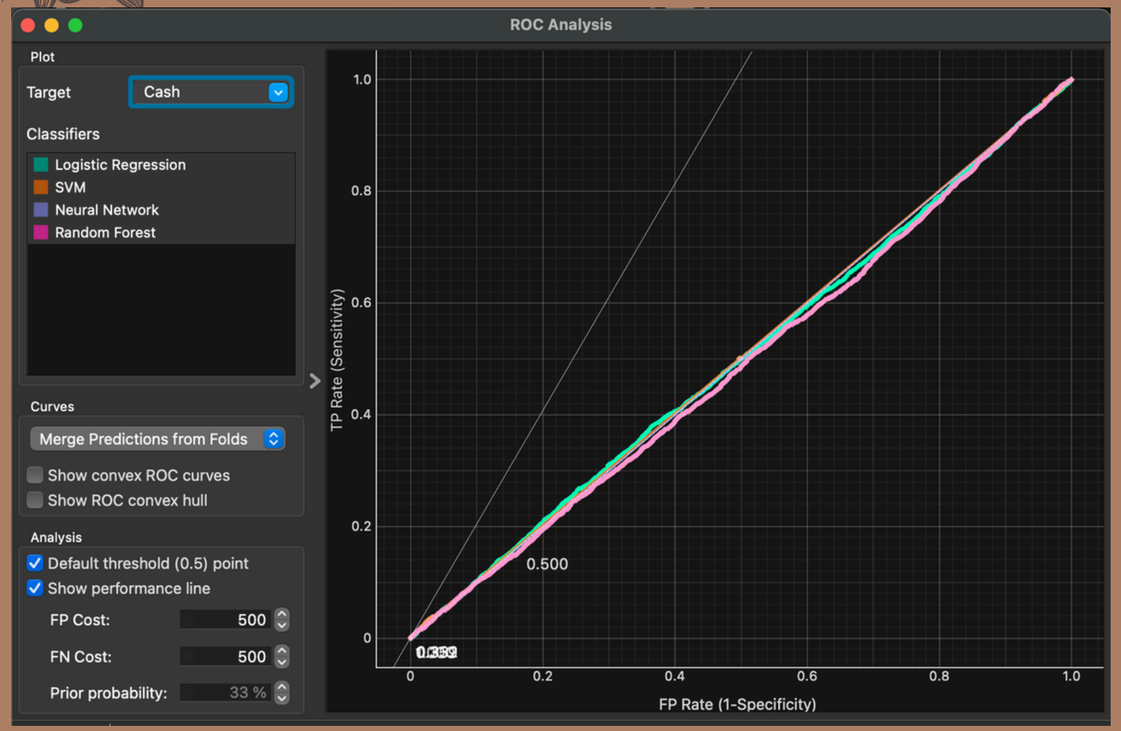
Models

- SVM
- Neural
- Network
- Logistic Regression
- Random Forest
- Gradient Boosting

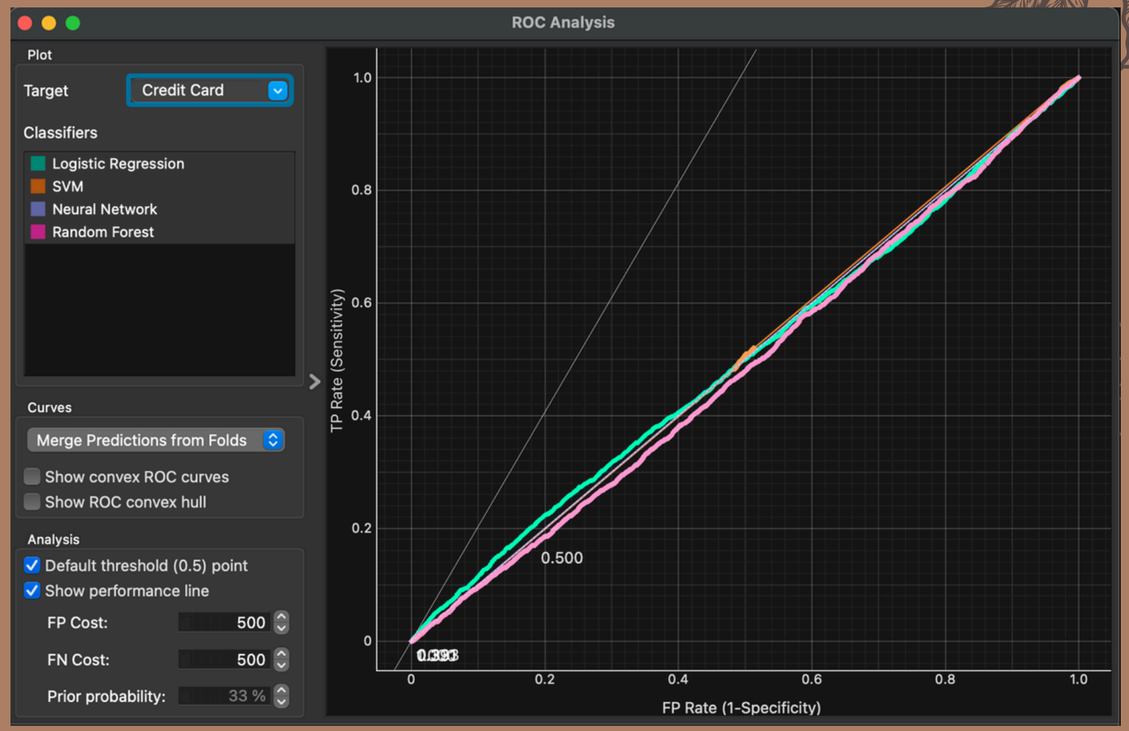
- MCC values are negative (~ -0.005 to -0.016), meaning the models have no meaningful correlation with actual payment methods.

Classification (ROC Analysis)

- Cash



- Credit Card



Similar Performance for Both "Cash" & "Credit Card"

- No significant difference in model performance across payment methods.
- Indicates that features do not provide enough discrimination.

Classification(Confusion Matrix)

Random Forest

Actual \ Predicted	Cash	Credit Card	Digital Wallet	Σ
Cash	1048	1079	1028	3155
Credit Card	1044	1037	1087	3168
Digital Wallet	1086	1076	1029	3191
Σ	3178	3192	3144	9514

Grafiant Boosting

Actual \ Predicted	Cash	Credit Card	Digital Wallet	Σ
Cash	807	1141	1207	3155
Credit Card	846	1068	1254	3168
Digital Wallet	797	1135	1259	3191
Σ	2450	3344	3720	9514

Logistic Regrssion

Actual \ Predicted	Cash	Credit Card	Digital Wallet	Σ
Cash	982	965	1208	3155
Credit Card	1031	978	1159	3168
Digital Wallet	1039	1040	1112	3191
Σ	3052	2983	3479	9514

SVM

Actual \ Predicted	Cash	Credit Card	Digital Wallet	Σ
Cash	2849	306	0	3155
Credit Card	2878	290	0	3168
Digital Wallet	2934	257	0	3191
Σ	8661	853	0	9514

Neural Network

Actual \ Predicted	Cash	Credit Card	Digital Wallet	Σ
Cash	977	995	1183	3155
Credit Card	975	984	1209	3168
Digital Wallet	947	1057	1187	3191
Σ	2899	3036	3579	9514

SVM Fails Completely

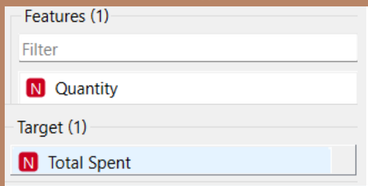
- Predicts almost everything as Cash, ignoring other payment methods.

Random Forest, Neural Network, Logistic Regression, and Gradient Boosting

- Distribute predictions evenly but incorrectly, meaning no strong pattern is learned.

Regression

Linear Regression



Model	MSE	RMSE	MAE	MAPE	R2
Linear Regression	17.902	4.231	3.177	0.530	0.512

The model has moderate predictive power (R^2 = 0.512) but a high error rate (MAPE = 53%), indicating room for improvement. The error metrics suggest the model struggles with accuracy.

Multiple Linear Regression

Model	MSE	RMSE	MAE	MAPE	R2
Linear Regression (1)	3.240	1.800	1.270	0.341	0.912

Polynomial Regression

Model	MSE	RMSE	MAE	MAPE	R2
Linear Regression (2)	3.243	1.801	1.273	0.341	0.912

Polynomial Regression(overfitting with ridge regression)

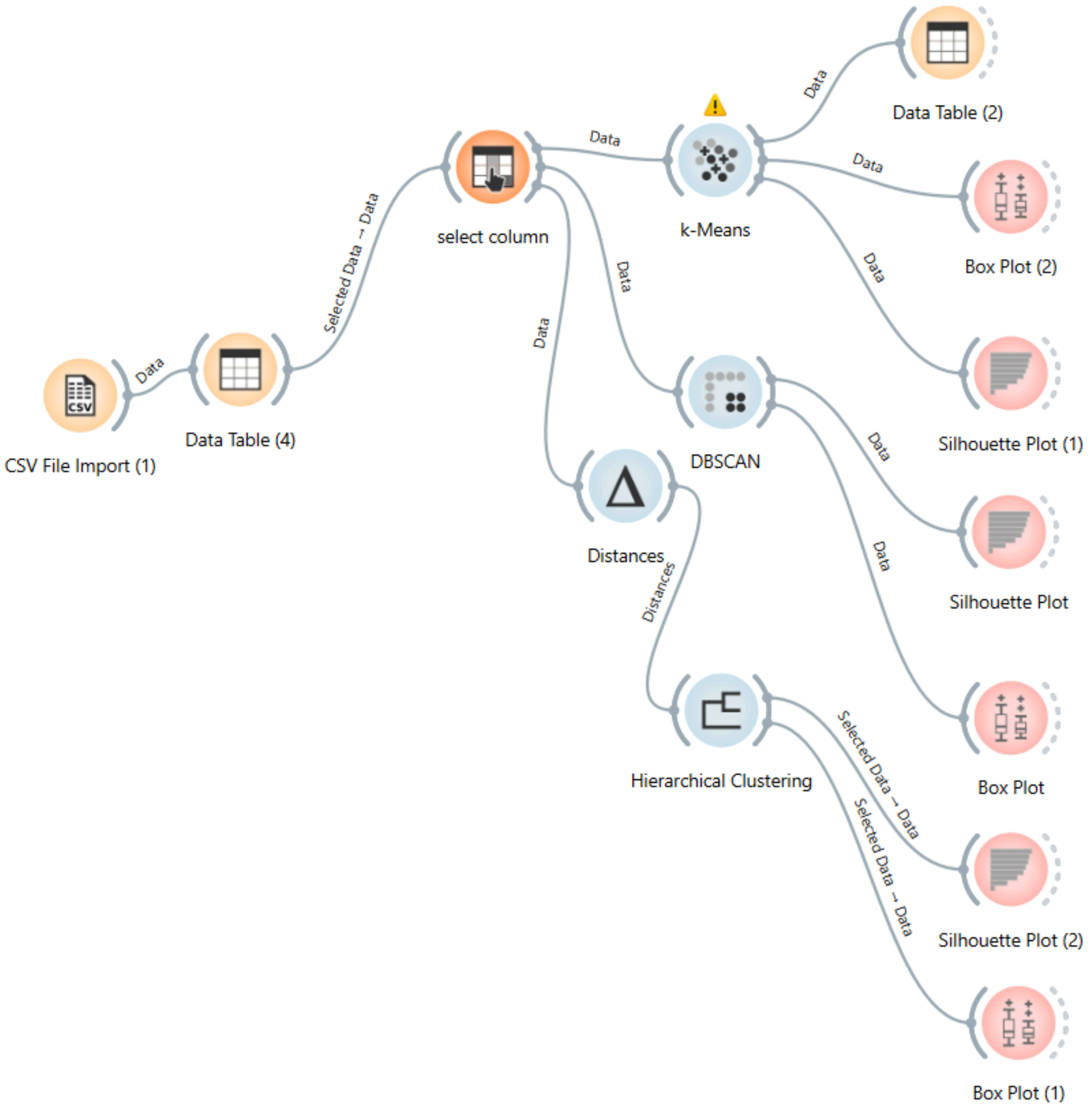
Model	MSE	RMSE	MAE	MAPE	R2
Linear Regression (3)	3.376	1.837	1.391	0.168	0.873

- Multiple Linear Regression and Polynomial Regression perform nearly identically (R^2 = 0.912), meaning polynomial transformation **didn't significantly improve accuracy.**
- Polynomial Regression with Ridge Regularization **reduces overfitting** (lower MAPE: 0.168) but **slightly sacrifices accuracy** (R^2 = 0.873).
- Choose Polynomial + Ridge for **better generalization** or stick with Multiple Linear Regression for **maximum accuracy**.

Clustering

- We analysis clustering to our dataset with K-Means, DBScan and Hierarchical Clustering

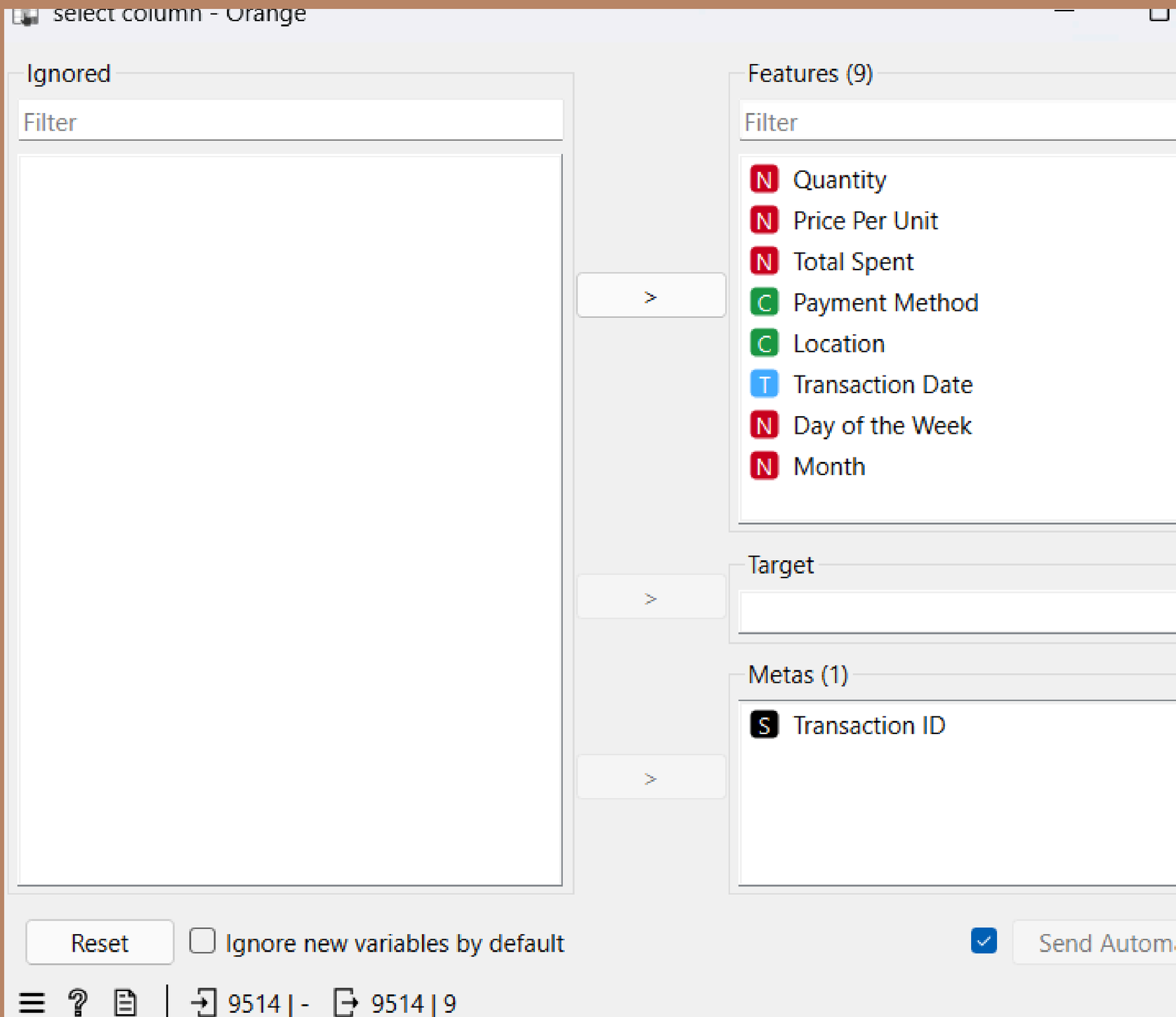
- This is our clustering analysis orange workflow



Clustering

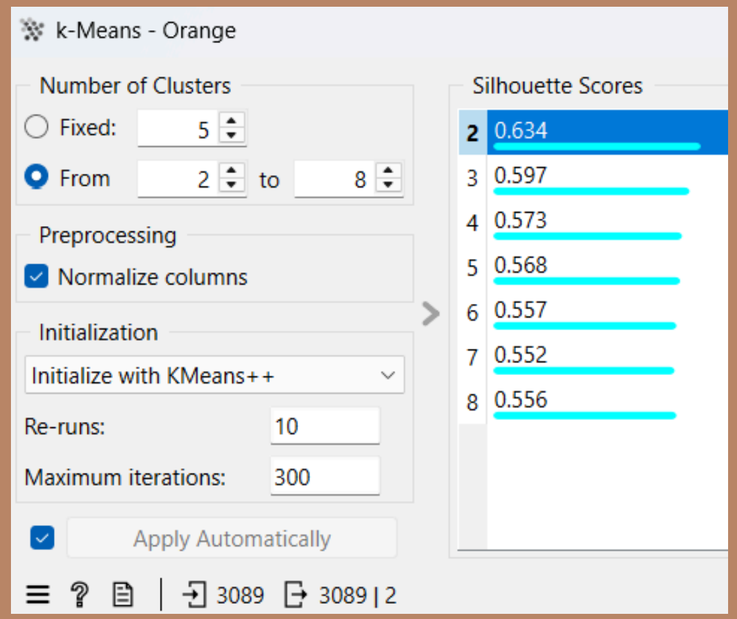
Select column

- In the Select column, we left the Target Values blank for clustering.



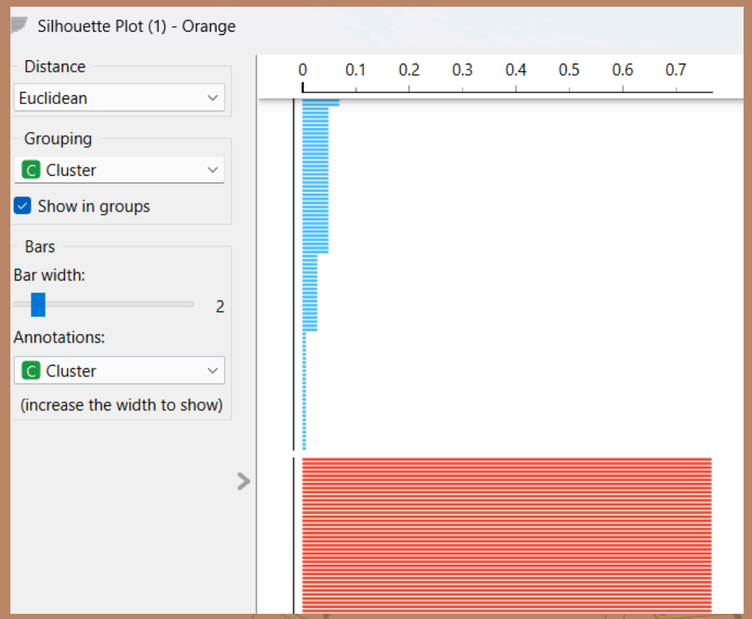
Clustering

K-Mean Clustering



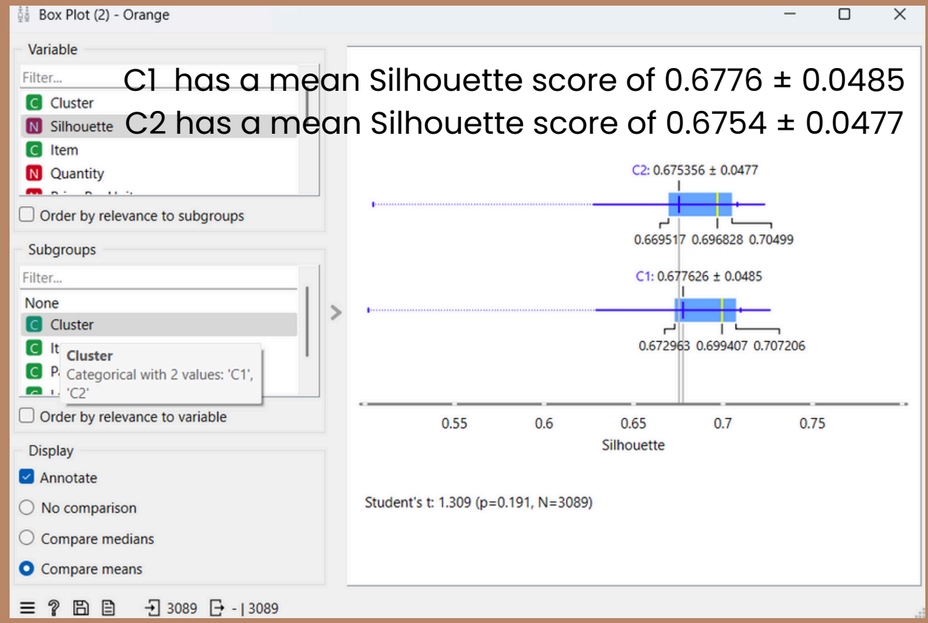
- In K-Means, when K=2, it is considered the best K-Means value because it has the highest values within the 2-8 cluster range.

Silhouette Plot



- In Silhouette plot, there is no misclassified data points.

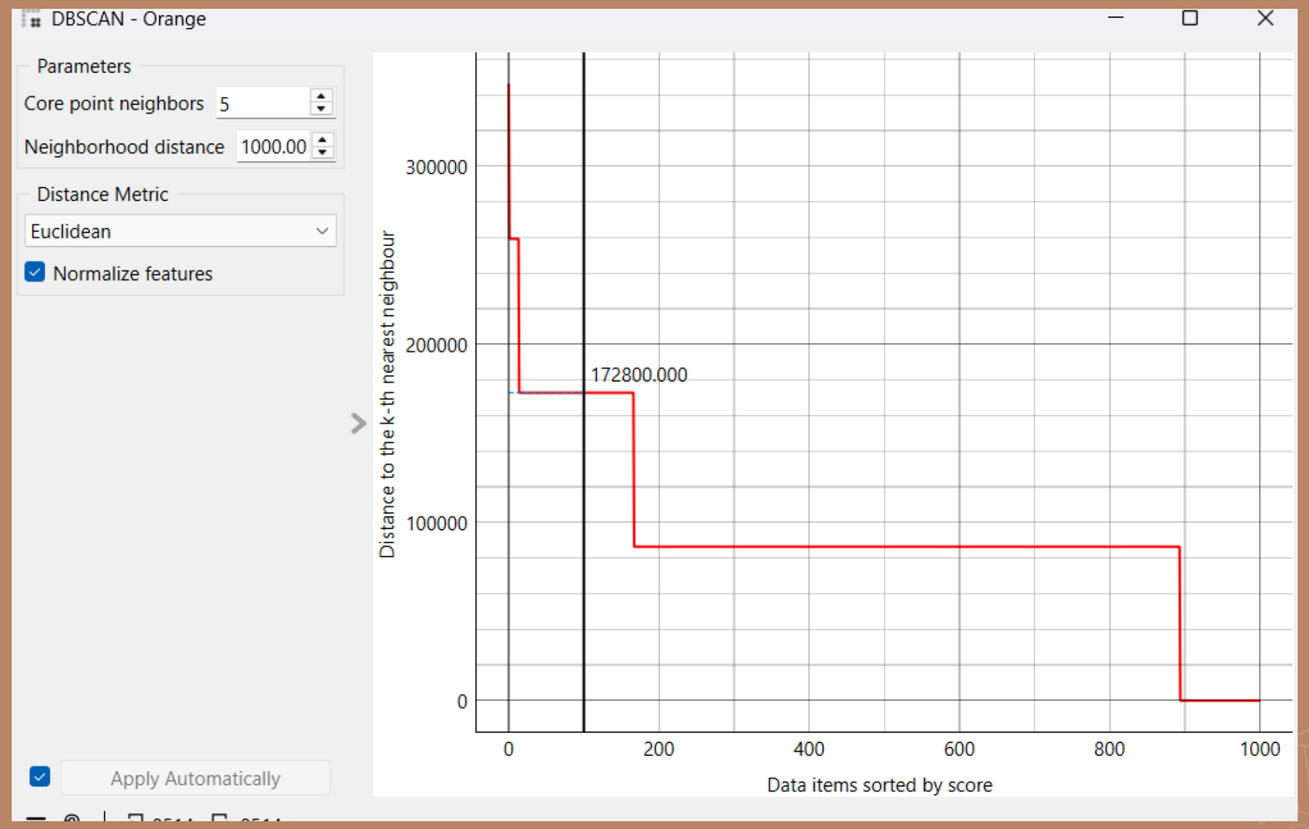
Box Plot



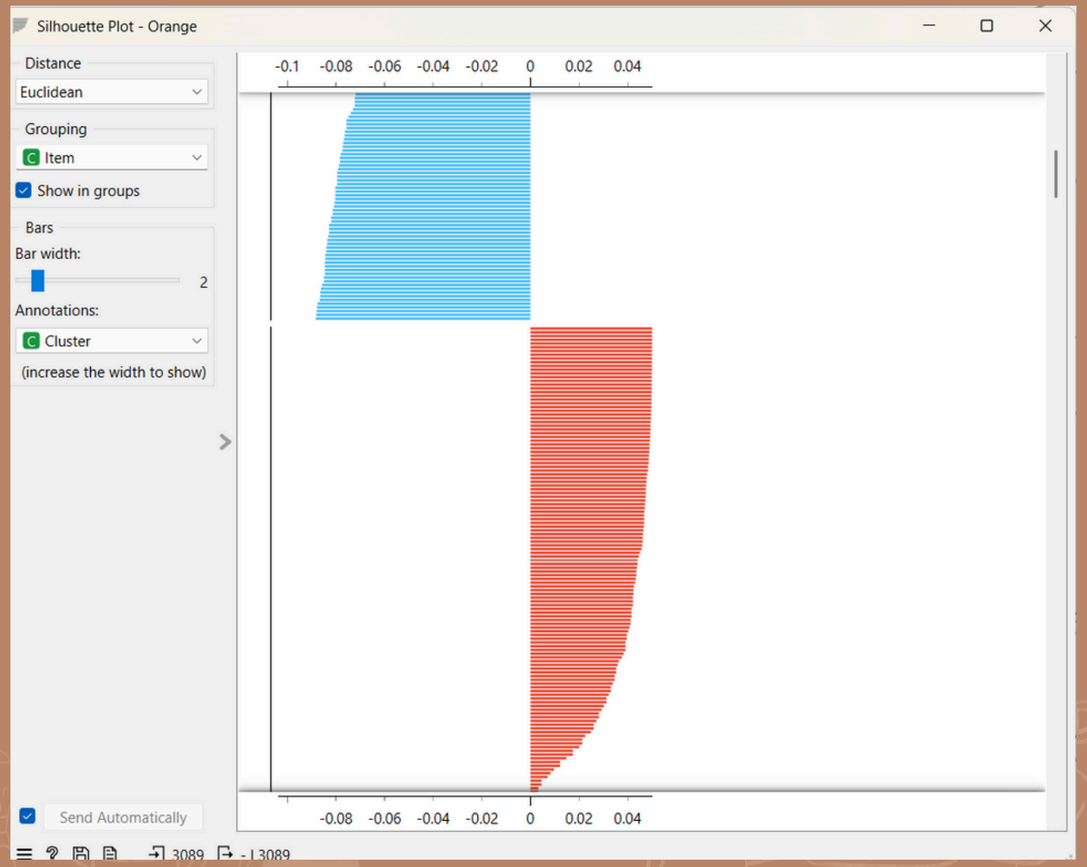
- In Box Plot, Silhouette scores are nearly the same for both clusters.
- Since p > 0.05, there is no strong evidence that the clustering quality is significantly different between the two groups.

Clustering

DB Scan



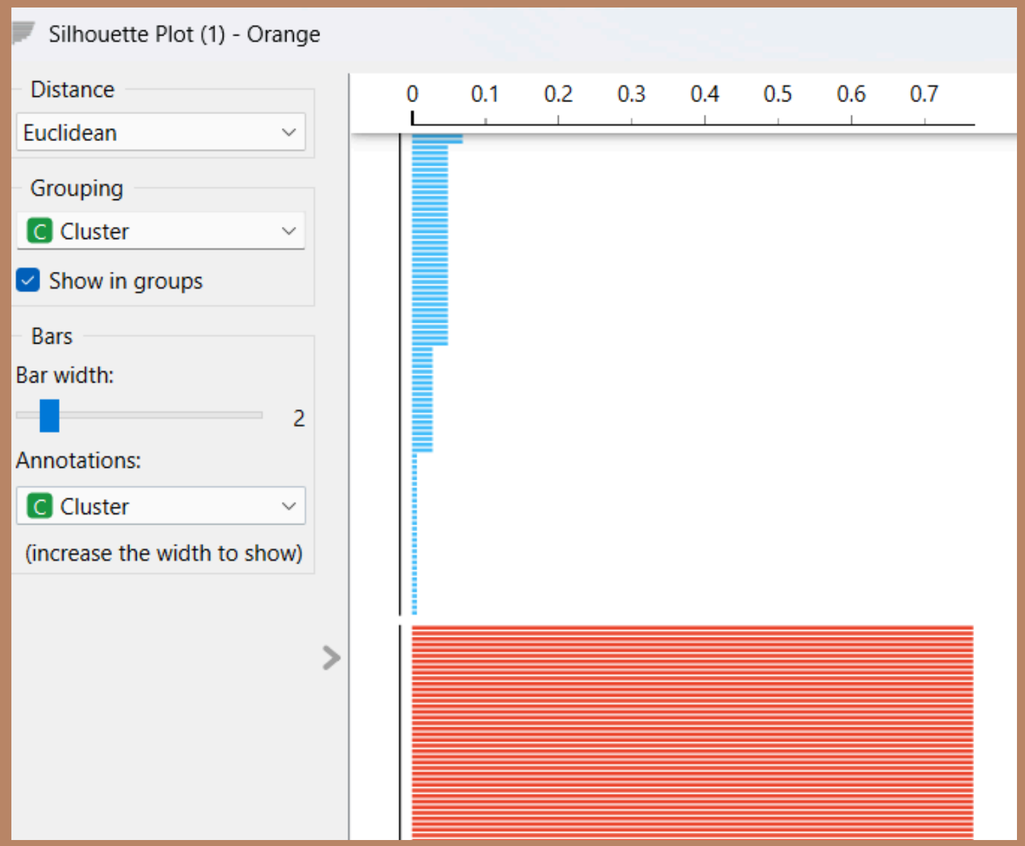
DB Scan Silhouette Plot



- In DBSCAN Silhouette Plot, we grouped with items and there are many misclassified data points in clusters cakes, coffee , and cookies.

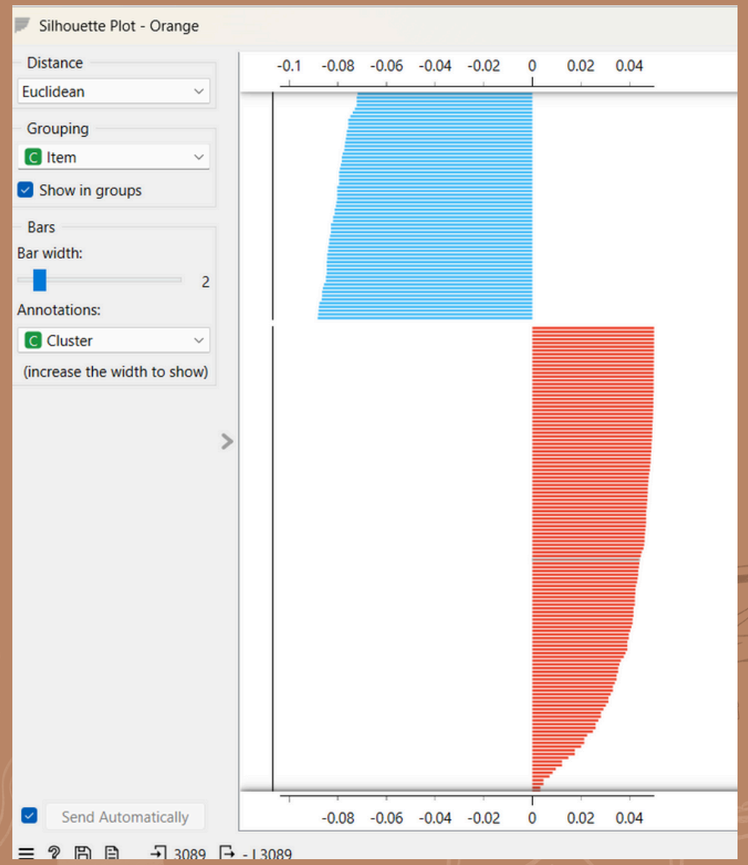
Clustering

K-Mean Silhouette Plot



VS

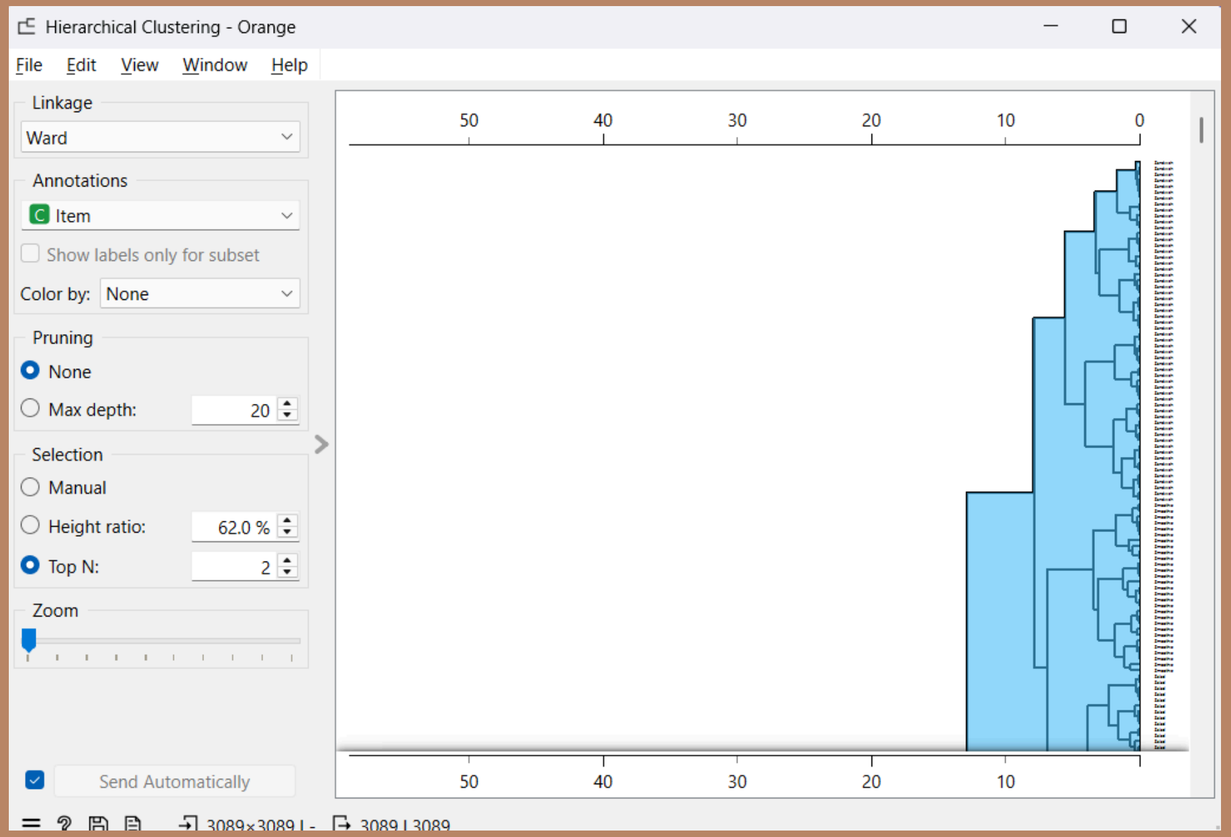
DB Scan Silhouette Plot



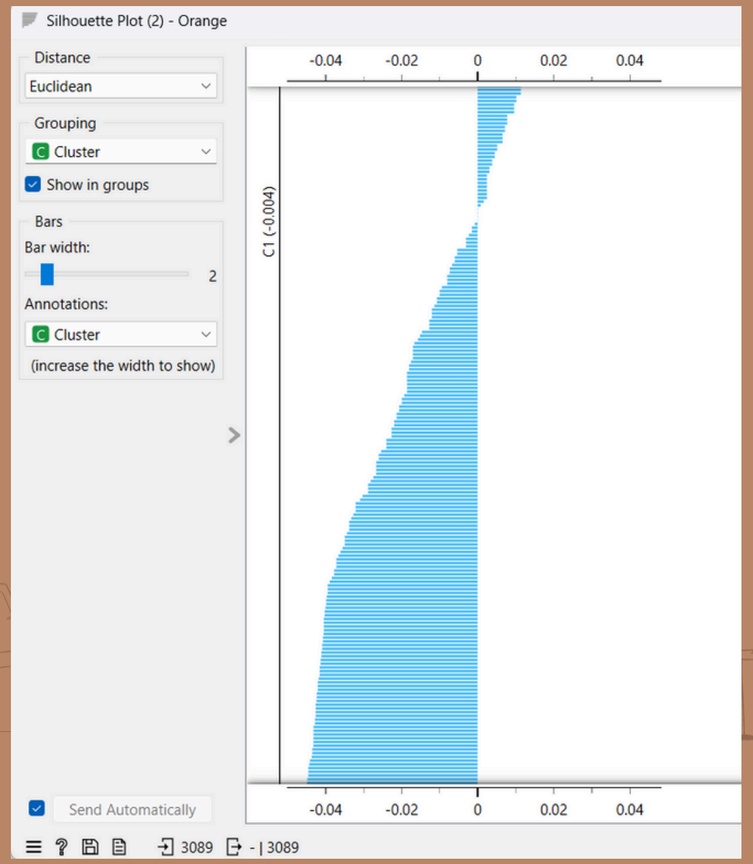
- So, Comparing with K-Mean Silhouette Plot and DB Scan Silhouette Plot, K-Means is better than DBSCAN as K-Mean has no misclassified data points.

Clustering

Hierarchical clustering



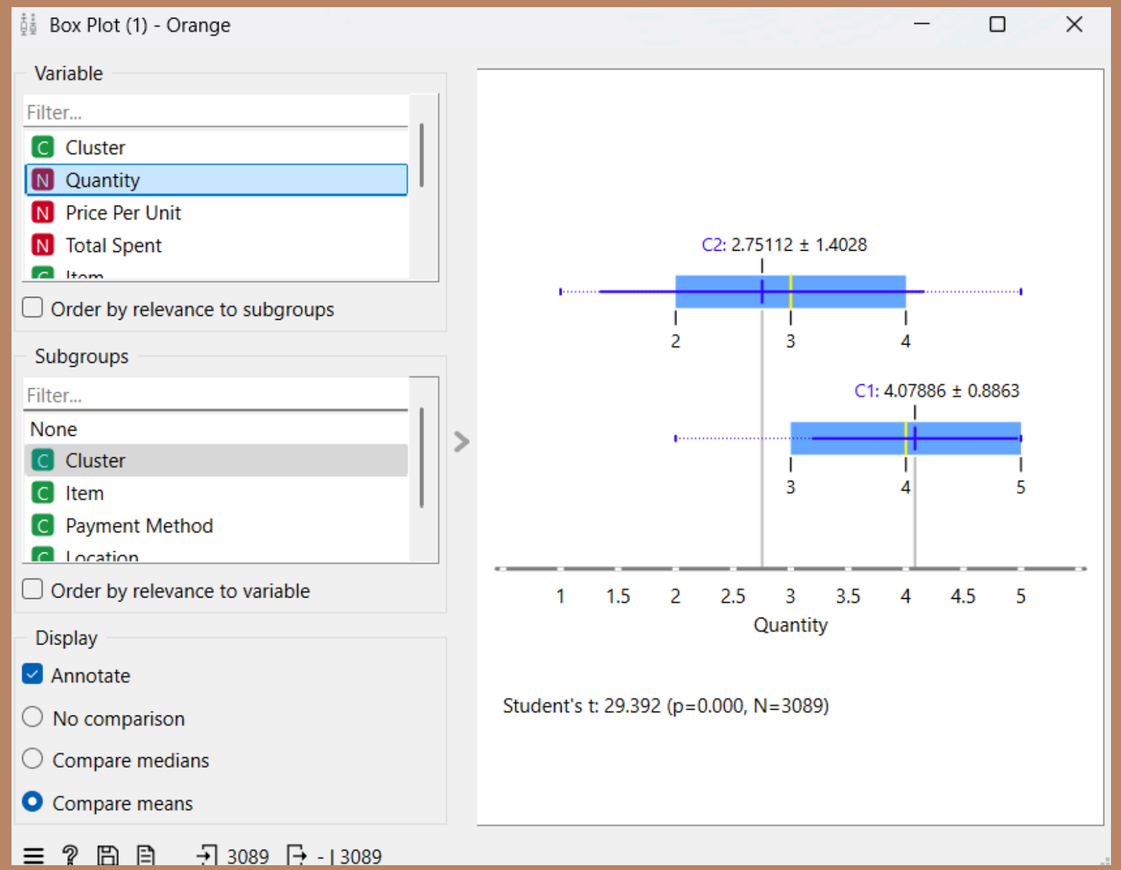
Silhouette Plot



In Hierarchical clustering Silhouette Plot , there are many misclassified data points.

Clustering

Hierarchical Clustering Box Plot



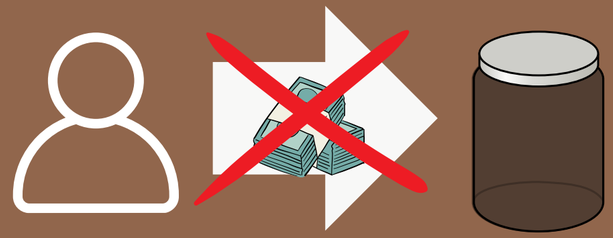
C1 (Cluster 1): Mean 4.07886 ± 0.8863
C2 (Cluster 2): Mean 2.75112 ± 1.4028

- C1 has a higher mean quantity (closer to 4) with a lower standard deviation (less spread, more consistency).
- C2 has a lower mean quantity (closer to 2.75) but with a higher standard deviation (more variation in quantity values).
- Both clusters have overlapping interquartile ranges, but C1 has a more concentrated range of values.
- Since $p < 0.05$, there is a statistically significant difference between the quantity distributions of C1 and C2.
- This means the clusters have distinct quantity characteristics, making the clustering separation meaningful.

Association & Text mining

Why we cannot perform text mining:

- The dataset lacks textual content like reviews or product descriptions
- Text mining requires large text fields for NLP techniques

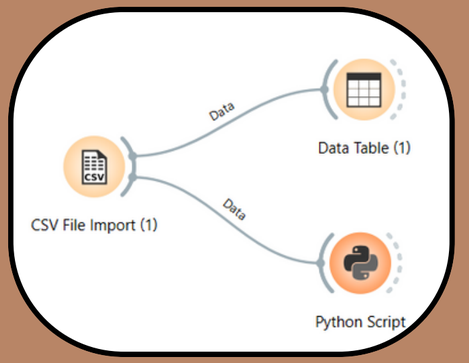


Why we cannot perform association rule mining:

- Association requires frequent item sets with strong relationships
- The café sales dataset lacks transactional pairs (e.g., customer A always buys item B with item C)

Python Script

- Automates data analysis for cafe sales trends.
- Generates visualized reports (bar charts + HTML tables).
- Helps in business decision-making using Orange.




```python
import pandas as pd
import os
import matplotlib.pyplot as plt
from Orange.data.pandas_compat import table_from_frame, table_to_frame
import warnings
warnings.simplefilter("ignore", category=DeprecationWarning)
```

- Import libraries

- **Data Visualization**
- Generates bar charts for easy interpretation using Matplotlib.

```python
# Function to generate and save visualizations
def save_plot(df, x, y, title, filename):
    plt.figure(figsize=(10, 5))
    plt.bar(df[x], df[y], color='skyblue')
    plt.xlabel(x)
    plt.ylabel(y)
    plt.title(title)
    plt.xticks(rotation=45)
    plt.tight_layout()
    plt.savefig(os.path.join(save_dir, filename))
    plt.close()SS
```

```python
# Function to style DataFrame and convert it to HTML format
def make_html_from_df(df, columns, cmap='Greens'):
    numeric_columns = df.select_dtypes(include=['number']).columns
    return (
        df
        .style
        .format({col: format_number for col in numeric_columns})
        .background_gradient(subset=columns, cmap=cmap)
        .set_table_styles([
            {'selector': 'table', 'props': [('border', '2px solid gray')]},
            {'selector': 'th, td', 'props': [('border', '1px solid gray'), ('padding', '8px')]},
            {'selector': 'th', 'props': [('background-color', '#f0f0f0'), ('font-weight', 'bold')]},
            {'selector': 'td', 'props': [('text-align', 'right')]},
            {'selector': 'td:hover', 'props': [('background-color', '#888888')]}
        ])
        .hide(axis="index")
        .to_html()
    )
```

- **Styled HTML Reports**
- Uses CSS styling and gradients to highlight important trends.
- Saves reports as HTML files with embedded charts.

Python Script

Report 1: Total Revenue by Item & Payment Method

```python
# Report 1: Total revenue by Item and Payment Method
result1 = (
    df.groupby(['Item', 'Payment Method'], observed=True)
    .agg(total_revenue=('Total Spent', 'sum'))
    .reset_index()
    .sort_values(by='total_revenue', ascending=False)
)
save_plot(result1, 'Item', 'total_revenue', 'Total Revenue by Item & Payment', 'report1.png')
```

- Groups sales data by Item and Payment Method.
- Calculates total revenue for each combination.
- Sorts results in descending order of revenue.

Report 2: Total Quantity Sold per Item by Location

```python
# Report 2: Total quantity sold per Item by Location
result2 = (
    df.groupby(['Item', 'Location'], observed=True)
    .agg(total_quantity=('Quantity', 'sum'))
    .reset_index()
    .sort_values(by='total_quantity', ascending=False)
)
save_plot(result2, 'Item', 'total_quantity', 'Total Quantity Sold per Location', 'report2.png')
```

- Groups data by Item and Location.
- Calculates total quantity sold.
- Sorts items from highest to lowest sales.

Report 3: Total Sales per Day of the Week

```python
# Report 3: Total sales per Day of the Week
result3 = (
    df.groupby(['Day of the Week'], observed=True)
    .agg(total_sales=('Total Spent', 'sum'))
    .reset_index()
    .sort_values(by='total_sales', ascending=False)
)
save_plot(result3, 'Day of the Week', 'total_sales', 'Total Sales by Day of Week', 'report3.png')
```

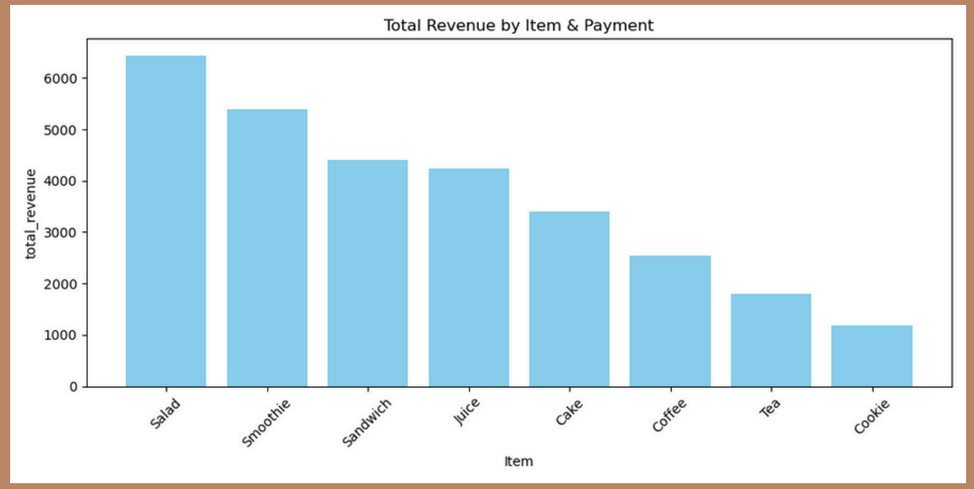
- Groups sales by Day of the Week.
- Calculates total sales per day.
- Sorts days from highest to lowest sales.

Report 4: Average Spending per Transaction per Month

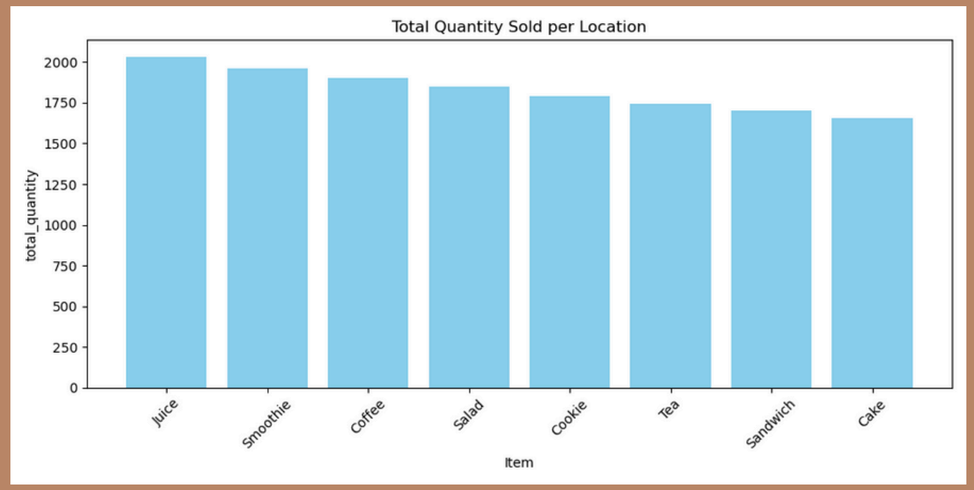
```python
# Report 4: Average spending per transaction per Month
result4 = (
    df.groupby(['Month'], observed=True)
    .agg(avg_spending=('Total Spent', 'mean'))
    .reset_index()
    .sort_values(by='avg_spending', ascending=False)
)
save_plot(result4, 'Month', 'avg_spending', 'Avg Spending per Month', 'report4.png')
```

- Groups data by Month.
- Calculates the average amount spent per transaction.
- Sorts months from highest to lowest spending.

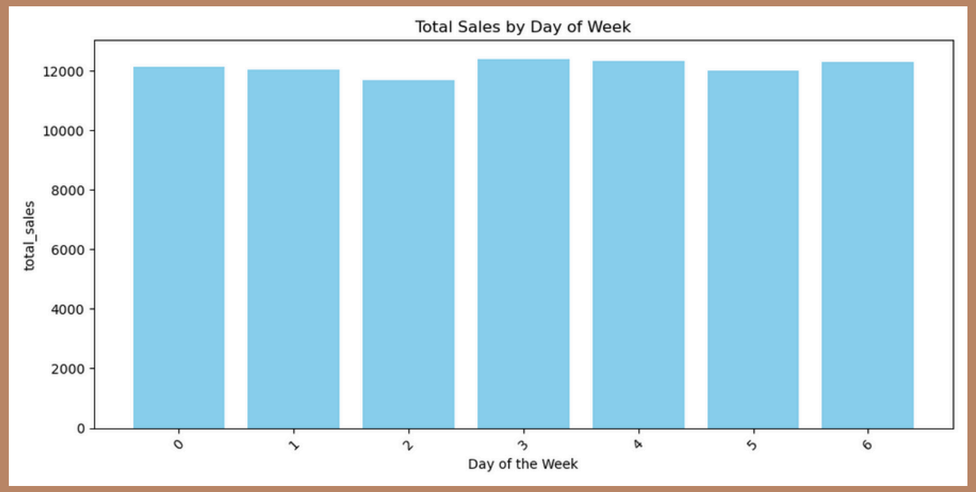
Python Script Outputs



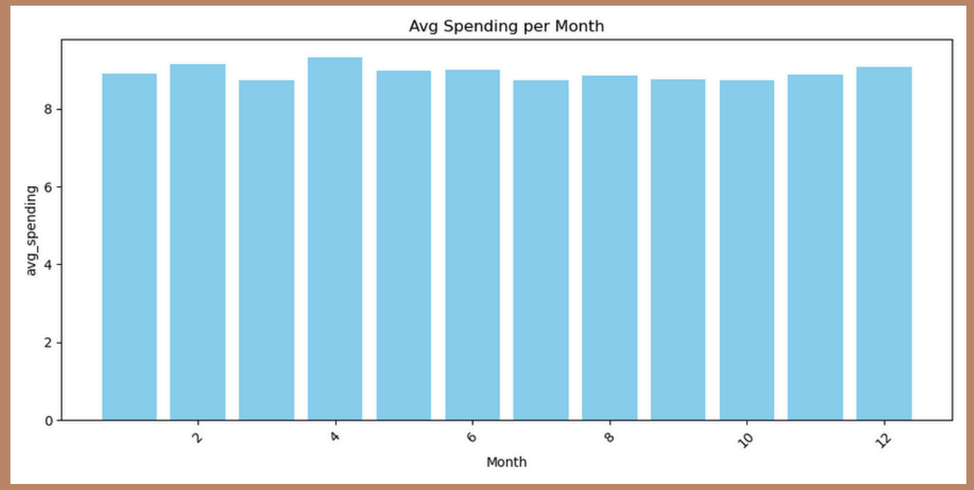
Report 1: Total Revenue by Item & Payment Method



Report 2: Total Quantity Sold per Item by Location



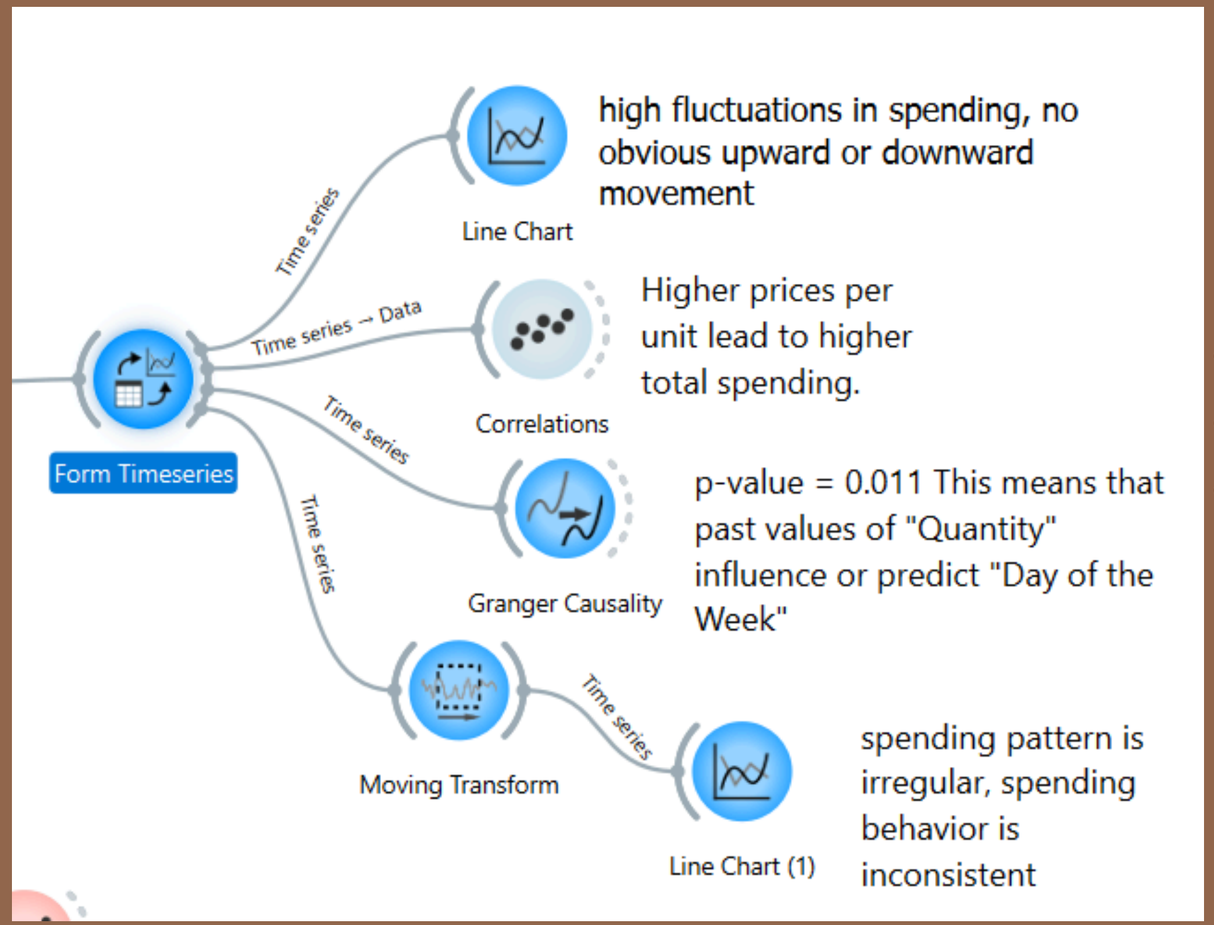
Report 3: Total Sales per Day of the Week



Report 4: Average Spending per Transaction per Month

Time Domain Analysis

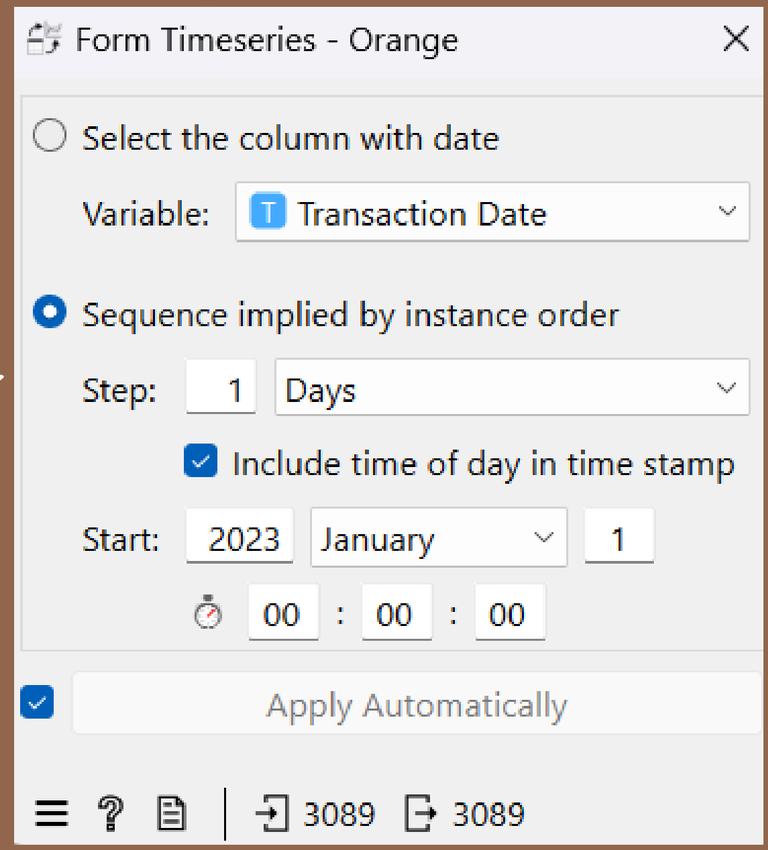
We observe the analysis with



Orange workflow

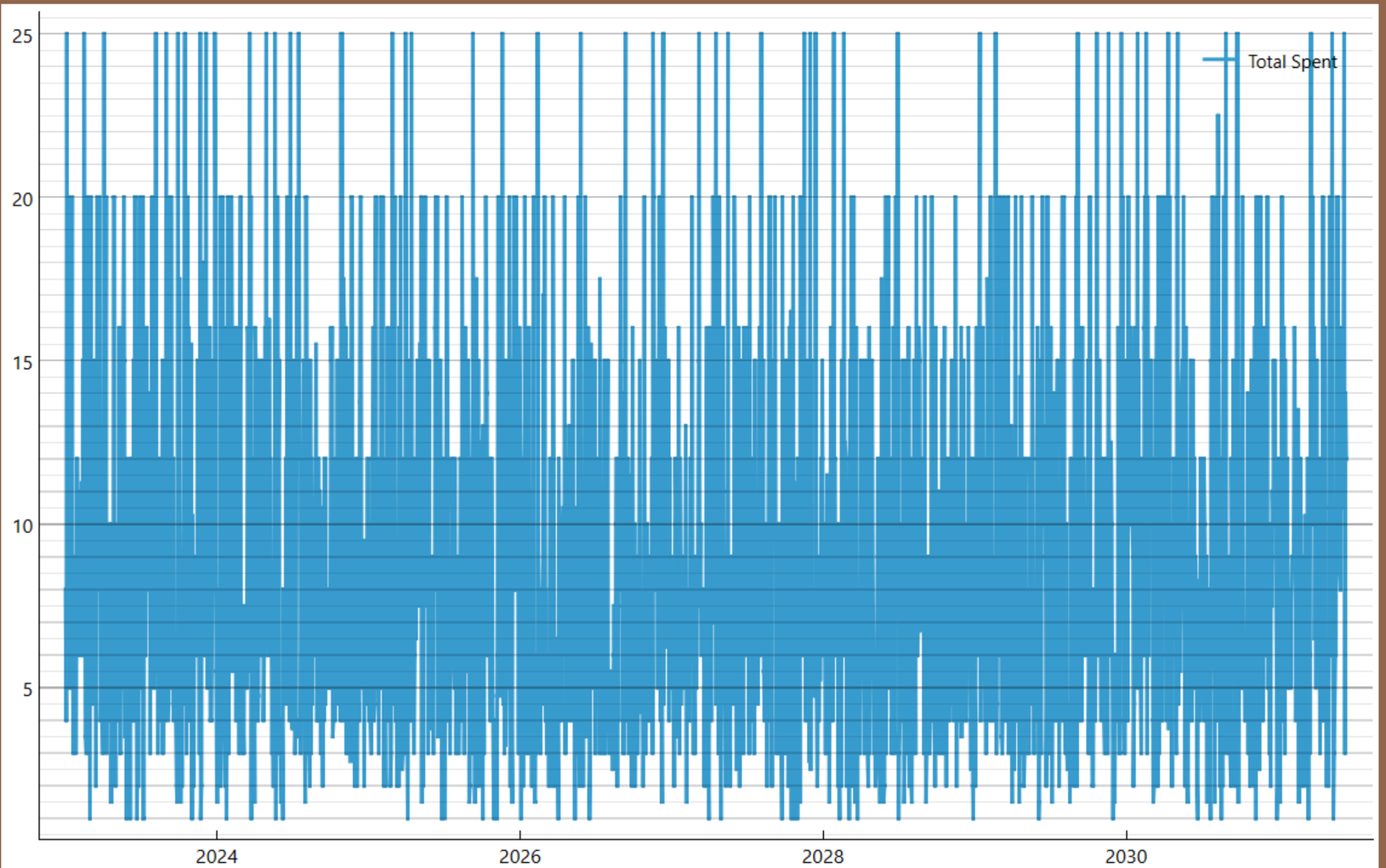
- **Line chart**
- **Correlations**
- **Granger Causality**

Form TImeseries setting is set to order by days. The time is set to start from 2023 January 1st



Time Domain Analysis

Line chart analysis report



Fluctuations in Total Spending:

- The graph exhibits significant daily variations in total spending, with frequent sharp peaks and drops.
- Spending ranges between low values (~5) and high values (~25), showing irregular spikes.

Data Anomalies:

- Some days show extreme peaks, meaning higher spending activity on certain occasions.
- There are also occasional low-spending days, which might indicate reduced customer activity.

Overall Spending Trend:

- The data does not indicate a clear upward or downward trend, suggesting stable but highly fluctuating spending behavior.

Time Domain Analysis

Correlations - Orange

Pearson correlation

(All combinations)

Filter ...

1	+0.695	Quantity	:	Total Spent
2	+0.607	Price Per Unit	:	Total Spent
3	-0.020	Quantity	:	T
4	-0.020	T	:	Transaction Date
5	-0.019	T	:	Total Spent
6	-0.018	Quantity	:	Transaction Date
7	-0.010	Total Spent	:	Transaction Date
8	-0.009	Price Per Unit	:	T
9	-0.008	Price Per Unit	:	Transaction Date
10	+0.001	Price Per Unit	:	Quantity

Correlation Analysis Report

Strongest Positive Correlations

- Quantity & Total Spent (+0.695)
- Price Per Unit & Total Spent (+0.607)

Weak or Negligible Correlations

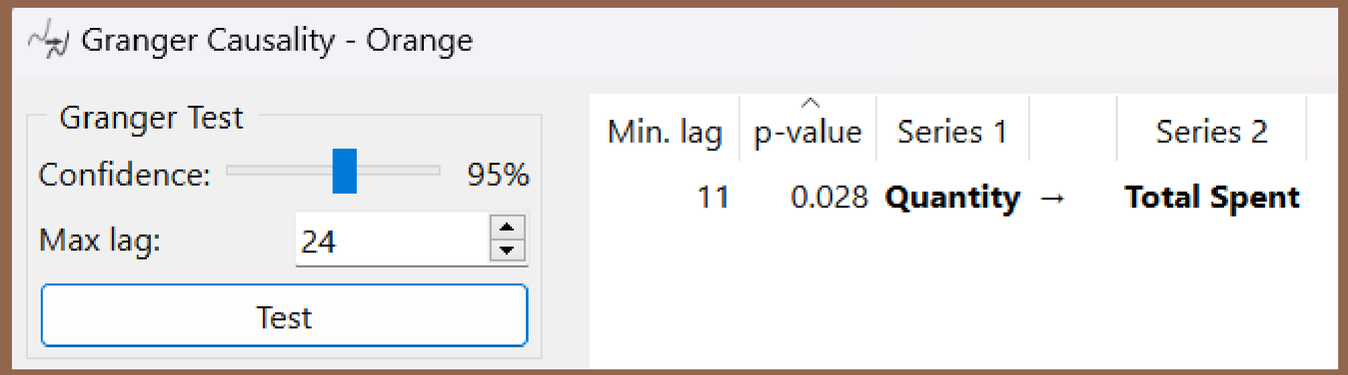
- Quantity & Transaction Date (-0.018)
- Total Spent & Transaction Date (-0.010)
- Price Per Unit & Transaction Date (-0.008)

Business Insight

- Leverage Quantity Discounts: Since higher quantity purchases lead to higher total spending, implementing bundle deals or discounts on bulk purchases could further boost revenue.
- The correlation analysis confirms that the biggest sales drivers in the café are the number of items sold and their price per unit.

Time Domain Analysis

Granger Causality Report



Causality Detected (Quantity → Total Spent)

- The test result shows that Quantity Granger-causes Total Spent with a p-value of 0.028.
- Since the p-value is below 0.05, we reject the null hypothesis and conclude that past values of Quantity have a statistically significant impact on predicting Total Spent.

Minimum Lag: 11

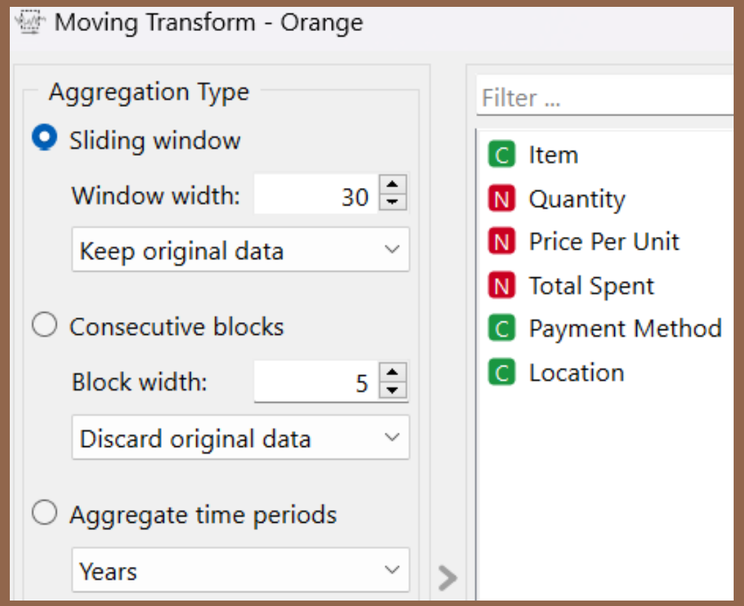
- The minimum lag of 11 suggests that the effect of quantity on total spending is observed after approximately 11 time steps (days, if daily data is used).
- This means changes in quantity affect total spending after about 11 days.

Business Insight

The Granger Causality test confirms that Quantity influences Total Spent, with a predictive lag of 11 time steps. Businesses can leverage this insight for forecasting revenue, adjusting stock levels, and implementing promotions to maximize profits.

Time Domain Analysis

Moving Transform and Line graph Analysis

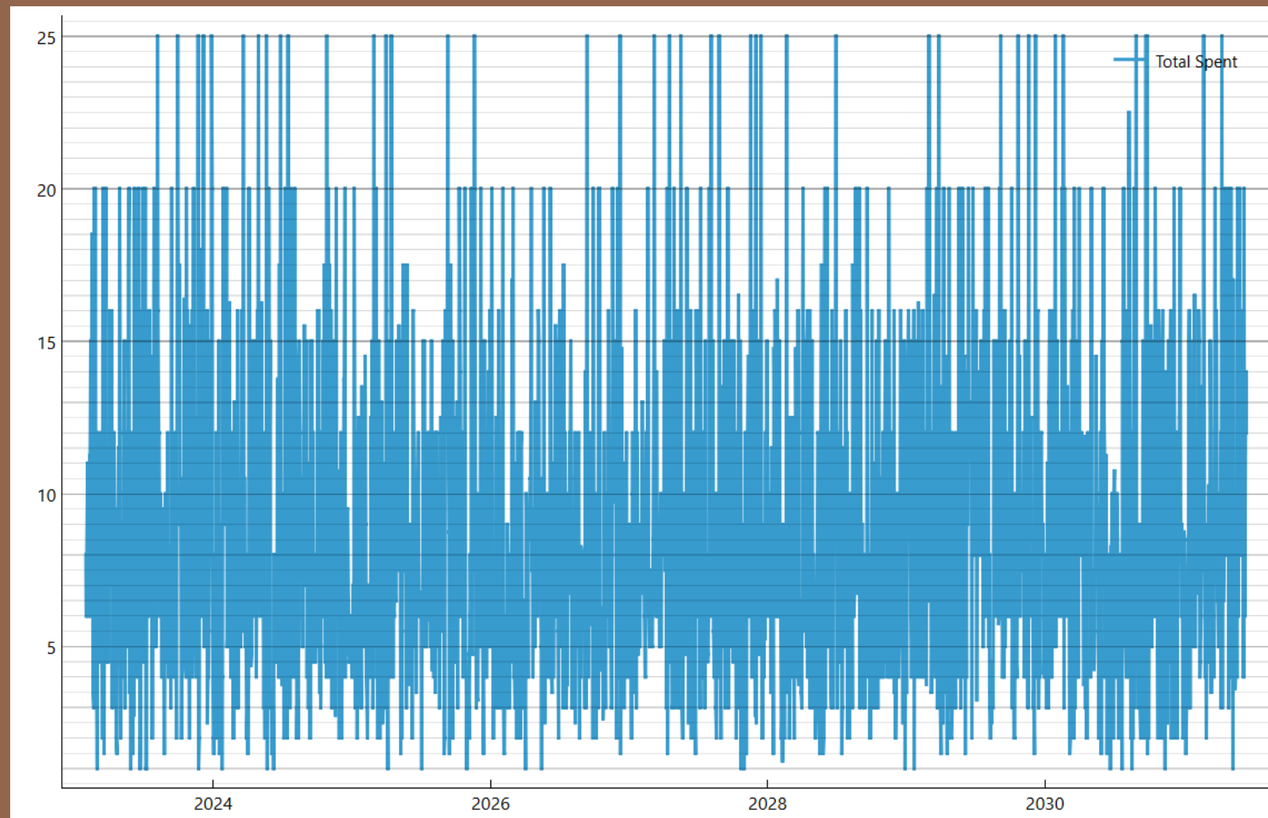


Smoothed Spending Trends

- **The 30-day moving window averages daily spending, reducing extreme fluctuations while maintaining overall trends.**

High Variability Despite Smoothing

- **Even after applying the moving window, Total Spent remains highly variable.**



The 30-day moving average transformation successfully reduces short-term noise and provides better visibility into customer spending trends. However, spending still appears highly variable, indicating the need for further aggregation or external factor analysis.

Thank you very much!