Name	Agata Gabara
Contact Number	+447904220617
Project Title (Example – Week1, Week2, Week3)	Week 4

VICTORIA SOLUTIONS
CONSULTING GROUP

Project Guidelines and Rules

1. **Formatting and Submission**
   - **Format:** Use a readable font (e.g., Arial/Times New Roman), size 12, 1.5 line spacing.
   - **Title:** Include Week and Title (Example - Week 1: TravelEase Case Study.)
   - **File Format:** Submit as PDF or Word file to contact@victoriasolutions.co.uk
   - **Page Limit:** 4–5 pages, including the title and references.

2. **Answer Requirements**
   - **Word Count:** Each answer should be 100–150 words; total 800–1,200 words.
   - **Clarity:** Write concise, structured answers with key points.
   - **Tone:** Use formal, professional language.

3. **Content Rules**
   - Answer all questions thoroughly, referencing case study concepts.
   - Use examples where possible (e.g., risk assessment techniques).
   - Break complex answers into bullet points or lists.

4. **Plagiarism Policy**
   - Submit original work; no copy-pasting.
   - Cite external material in a consistent format (e.g., APA, MLA).

5. **Evaluation Criteria**
   - **Understanding:** Clear grasp of business analysis principles.
   - **Application:** Effective use of concepts like cost-benefit analysis and Agile/Waterfall.
   - **Clarity:** Logical, well-structured responses.
   - **Creativity:** Innovative problem-solving and examples.
   - **Completeness:** Answer all questions within the word limit.

6. **Deadlines and Late Submissions**
    - **Deadline:** Submit on time; trainees who submit fail to submit the project will miss the “Certificate of Excellence”

7. **Additional Resources**
    - Refer to lecture notes and recommended readings.
    - Contact the instructor or peers for clarifications before the deadline.

START YOUR PROJECT FROM HERE:

1.Cleaned dataset.

Summary:

The dataset consists of 405 rows and 14 columns, capturing a broad range of customer-related information. It includes customer demographics such as Customer_ID, Age, Gender, and Income, as well as financial attributes like Credit_Score, Loan_Amount, and Previous_Defaults. Marketing and sales data are also incorporated, covering Marketing_Spend, Purchase_Frequency, Seasonality, and Sales, while behavioral and outcome variables such as Spending_Score, Customer_Churn, and Defaulted help in understanding customer behaviour and business risk. This structure provides a holistic view of customer profiles, their financial standing, and interactions with marketing and sales efforts.

From the aggregated insights, the average customer age is around 44 years, with an average income of about $84,300. Customers have an average credit score of ~573 and take out loans averaging $28,900, while their spending score averages around 51. The data shows a churn rate of 24.2%, but notably, no customers defaulted on their loans. The gender distribution is balanced, with 54% male and 46% female customers. Seasonality effects are evenly spread, with 34.6% in Medium, 33.8% in High, and 31.6% in Low categories, suggesting relatively balanced customer activity across different seasonal cycles.

I have used Python to clean dataset. Details are below.

project4.py

```python
import pandas as pd

df = pd.read_csv(r"C:\Users\PC\OneDrive\Desktop\data.csv.csv")

print("Before filling missing values:")
print(df.isna().sum())    # counts of NaN values per column

# Fill NaN with mean
df.fillna(df.mean(numeric_only=True), inplace=True)

print("\nAfter filling missing values:")
print(df.isna().sum())    # confirm missing values are gone

# Preview first 5 rows of cleaned data
print("\nPreview of cleaned data:")
print(df.head())

# Save cleaned file
df.to_csv(r"C:\Users\PC\OneDrive\Desktop\cleaned_data.csv", index=False)
print("\n✅ Cleaned file saved to Desktop as 'cleaned_data.csv'")

```

Running: project4.py

```
Before filling missing values:
Customer_ID            0
Age                    0
Gender                 0
Income                50
Spending_Score         0
Credit_Score          50
Loan_Amount           50
Previous_Defaults      0
Marketing_Spend        0
Purchase_Frequency     0
Seasonality            0
Sales                  0
Customer_Churn         0
Defaulted              0
dtype: int64

After filling missing values:
Customer_ID           0
Age                   0
Gender                0
Income                0
Spending_Score        0
Credit_Score          0
Loan_Amount           0
Previous_Defaults     0
Marketing_Spend       0
Purchase_Frequency    0
Seasonality           0
Sales                 0
Customer_Churn        0
Defaulted             0
dtype: int64

Preview of cleaned data:
   Customer_ID  Age  Gender  ...  Sales  Customer_Churn  Defaulted
0            1   56  Female  ...  32526               0          0
1            2   69    Male  ...  78493               0          0
2            3   46    Male  ...  57198               1          0
3            4   32  Female  ...  48395               0          0
4            5   60    Male  ...  29031               1          0

[5 rows x 14 columns]

✅ Cleaned file saved to Desktop as 'cleaned_data.csv'
>>>
```

The script loads a CSV file, checks for missing values, and fills numeric NaNs with the column mean. It then previews the cleaned data and saves it as a new CSV file on the Desktop.

```python
import pandas as pd
import numpy as np
from pathlib import Path

input_file = Path(r"C:\Users\PC\OneDrive\Desktop\data.csv.csv")
output_file = Path(r"C:\Users\PC\OneDrive\Desktop\cleaned_data.csv")
df = pd.read_csv(input_file)
print("🔎 Missing values before cleaning:")
print(df.isna().sum(), "\n")
df.fillna(df.mean(numeric_only=True), inplace=True)
print("✅ Missing values after cleaning:")
print(df.isna().sum(), "\n")

Q1 = df.quantile(0.25, numeric_only=True)
Q3 = df.quantile(0.75, numeric_only=True)
IQR = Q3 - Q1
df = df[~((df.select_dtypes(include=[np.number]) < (Q1 - 1.5 * IQR)) |
          (df.select_dtypes(include=[np.number]) > (Q3 + 1.5 * IQR))).any(axis=1)]

print(f"📊 Dataset shape after outlier removal: {df.shape}\n")
print("Preview of cleaned data:")
print(df.head(), "\n")
df.to_csv(output_file, index=False)
print(f"💾 Cleaned file saved to: {output_file}")

```

```
🔎 Missing values before cleaning:
Customer_ID            0
Age                    0
Gender                 0
Income                50
Spending_Score         0
Credit_Score          50
Loan_Amount           50
Previous_Defaults      0
Marketing_Spend        0
Purchase_Frequency     0
Seasonality            0
Sales                  0
Customer_Churn         0
Defaulted              0
dtype: int64
```

```
✅ Missing values after cleaning:
Customer_ID          0
Age                  0
Gender               0
Income               0
Spending_Score       0
Credit_Score         0
Loan_Amount          0
Previous_Defaults    0
Marketing_Spend      0
Purchase_Frequency   0
Seasonality          0
Sales                0
Customer_Churn       0
Defaulted            0
dtype: int64

📊 Dataset shape after outlier removal: (405, 14)

Preview of cleaned data:
   Customer_ID  Age  Gender  ...  Sales  Customer_Churn  Defaulted
0            1   56  Female  ...  32526               0          0
1            2   69    Male  ...  78493               0          0
2            3   46    Male  ...  57198               1          0
3            4   32  Female  ...  48395               0          0
4            5   60    Male  ...  29031               1          0

[5 rows x 14 columns]

💾 Cleaned file saved to: C:\Users\PC\OneDrive\Desktop\cleaned_data.csv
```

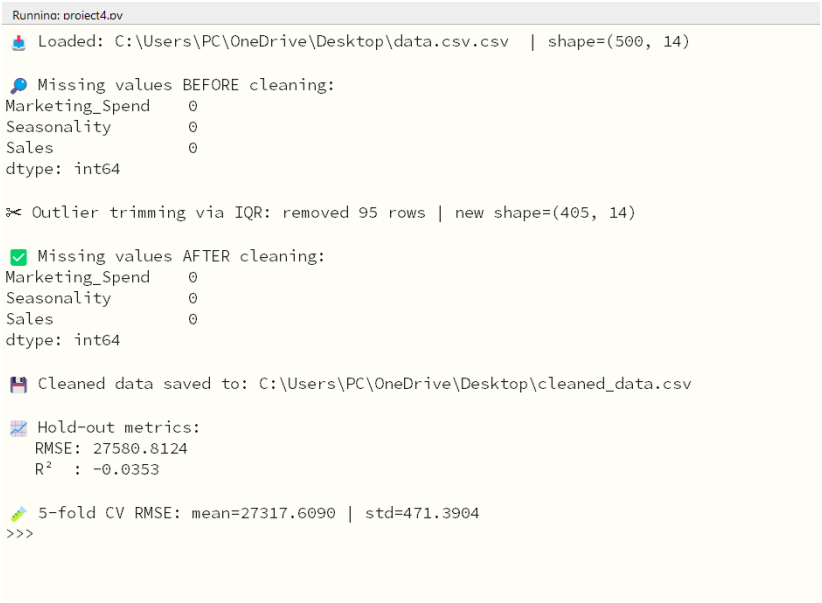
The script cleans the dataset by filling missing numeric values with column means and then removes outliers using the IQR method. Finally, it previews the cleaned data and saves it as a new CSV file on the Desktop.

```python
import sys
from pathlib import Path
import numpy as np
import pandas as pd
from sklearn.compose import ColumnTransformer
from sklearn.model_selection import train_test_split, KFold, cross_val_score
from sklearn.preprocessing import OneHotEncoder
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.pipeline import Pipeline
INPUT_FILE  = Path(r"C:\Users\PC\OneDrive\Desktop\data.csv.csv")
OUTPUT_FILE = Path(r"C:\Users\PC\OneDrive\Desktop\cleaned_data.csv")
TARGET = "Sales"
FEATURES = ["Marketing_Spend", "Seasonality"]   # adjust if needed
IQR_TRIM = True                                  # set False to disable outlier trimming

if not INPUT_FILE.exists():
    sys.exit(f"❌ Input file not found: {INPUT_FILE}")

df = pd.read_csv(INPUT_FILE)
print(f"📥 Loaded: {INPUT_FILE}  | shape={df.shape}")

missing_cols = [c for c in [TARGET, *FEATURES] if c not in df.columns]
if missing_cols:
    sys.exit(f"❌ Missing required column(s): {missing_cols}")

print("\n🔎 Missing values BEFORE cleaning:")
print(df[[*FEATURES, TARGET]].isna().sum())
```

```
num_cols = df.select_dtypes(include=[np.number]).columns.tolist()
df[num_cols] = df[num_cols].apply(pd.to_numeric, errors="coerce")
df[num_cols] = df[num_cols].fillna(df[num_cols].mean())

if IQR_TRIM:
    Q1 = df[num_cols].quantile(0.25)
    Q3 = df[num_cols].quantile(0.75)
    IQR = Q3 - Q1
    lower = Q1 - 1.5 * IQR
    upper = Q3 + 1.5 * IQR

    before = len(df)
    mask_outlier = ((df[num_cols] < lower) | (df[num_cols] > upper)).any(axis=1)
    df = df.loc[~mask_outlier].copy()
    after = len(df)
    print(f"\n✂️ Outlier trimming via IQR: removed {before - after} rows | new shape={df.shape}")

print("\n✅ Missing values AFTER cleaning:")
print(df[[*FEATURES, TARGET]].isna().sum())

df.to_csv(OUTPUT_FILE, index=False)
print(f"\n💾 Cleaned data saved to: {OUTPUT_FILE}")

X = df[FEATURES].copy()
y = df[TARGET].copy()

X_num = X.select_dtypes(include=[np.number]).columns.tolist()
```

```
X_cat = [c for c in X.columns if c not in X_num]

preprocessor = ColumnTransformer(
    transformers=[
        ("num", "passthrough", X_num),
        ("cat", OneHotEncoder(handle_unknown="ignore"), X_cat),
    ],
    remainder="drop",
)
model = LinearRegression()
pipe = Pipeline(steps=[
    ("prep", preprocessor),
    ("model", model),
])

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.20, random_state=42
)
pipe.fit(X_train, y_train)
pred = pipe.predict(X_test)
mse  = mean_squared_error(y_test, pred)
rmse = np.sqrt(mse)
r2   = r2_score(y_test, pred)
print(f"\n📈 Hold-out metrics:")
print(f"   RMSE: {rmse:.4f}")
print(f"   R²  : {r2:.4f}")
```

```
cv = KFold(n_splits=5, shuffle=True, random_state=42)
cv_mse = cross_val_score(pipe, X, y, scoring="neg_mean_squared_error", cv=cv)
cv_rmse = np.sqrt(-cv_mse)
print(f"\n🧪 5-fold CV RMSE: mean={cv_rmse.mean():.4f} | std={cv_rmse.std():.4f}")
```



Interpretation:

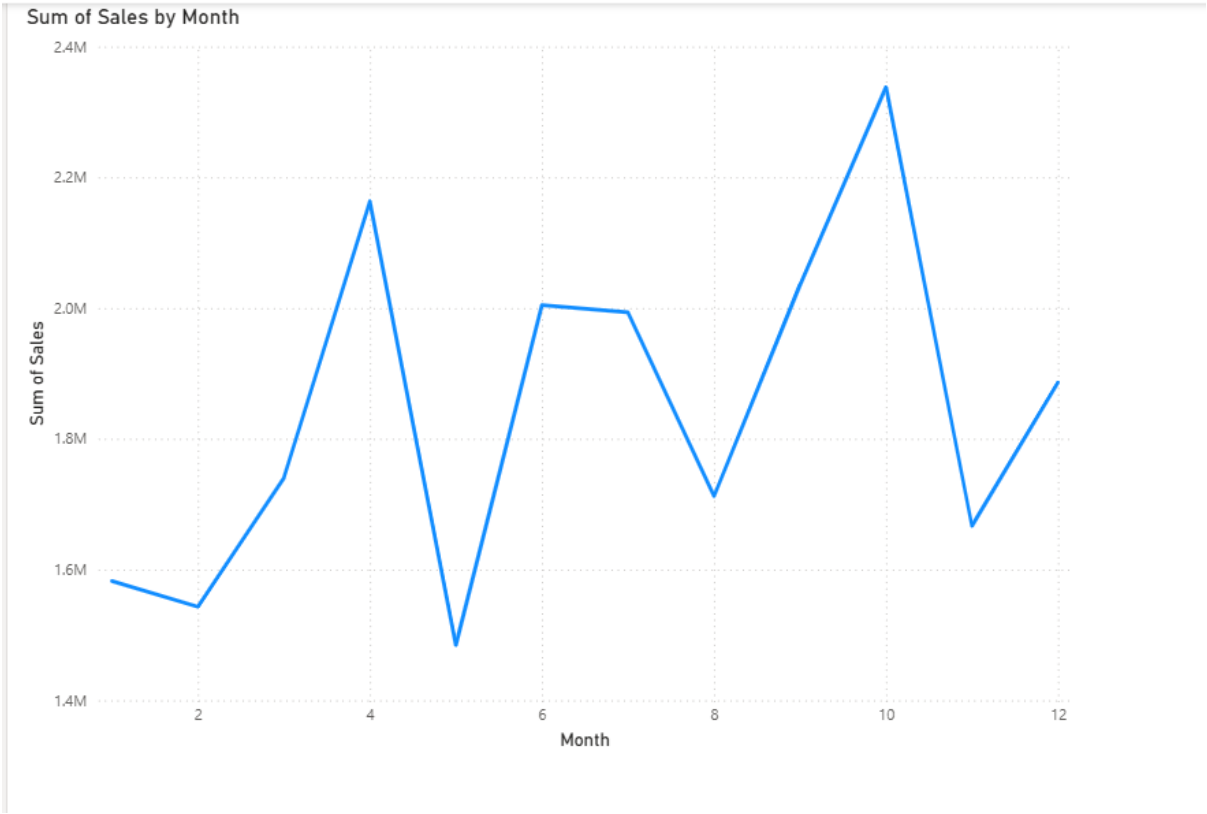
The dataset originally had 500 rows and 14 columns, with no missing values in the key features. After applying IQR-based trimming, 95 outliers were removed, leaving 405 rows. The linear regression model performed poorly on the hold-out test set (negative R^2, RMSE ≈ 27.6k), indicating it cannot explain the variance in Sales well. However, the 5-fold CV RMSE is consistent (~27.3k), suggesting the model is stable but underfits the data.

2.AI-powered visualizations.

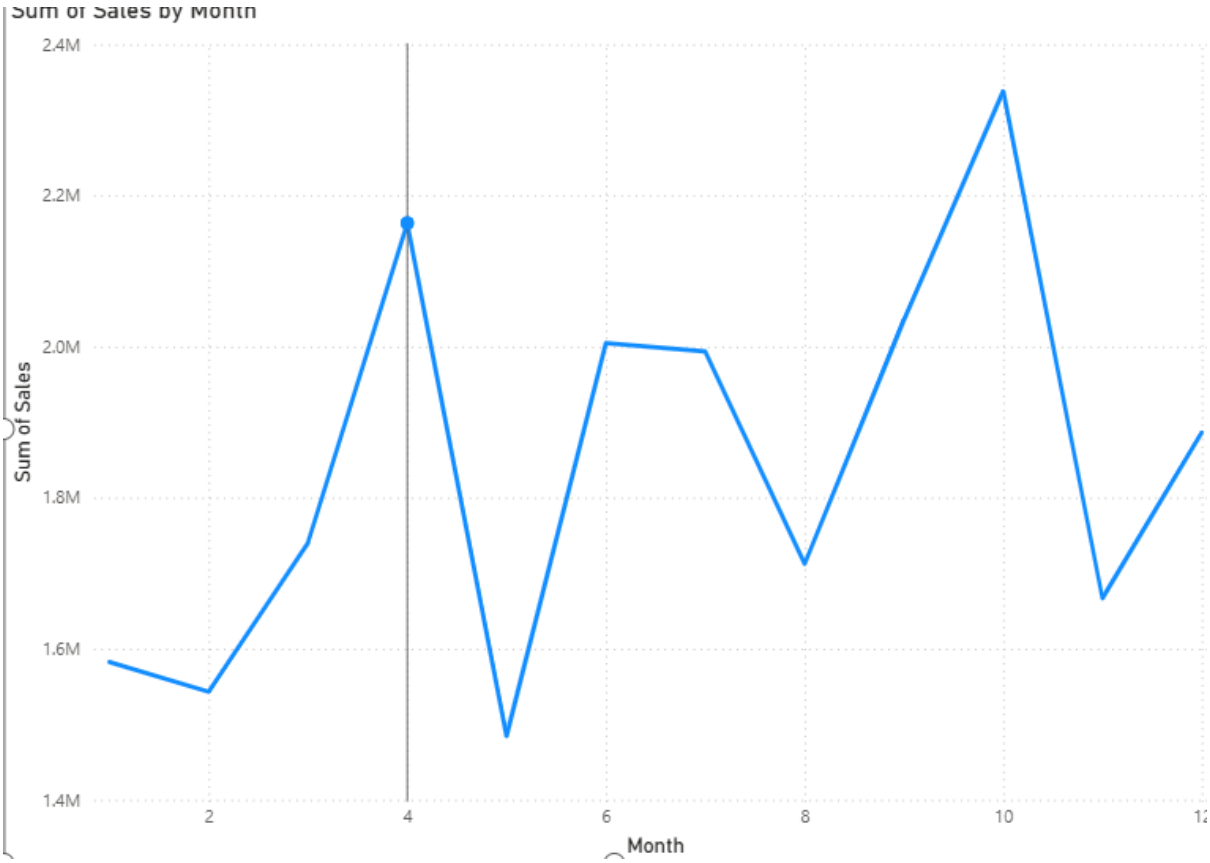
I have used POWER BI Desktop as I don't have any work or uni account to use the online version.

I have added a month column with DAX to the data set and I was able to create a line chart.

```
DAX

Month = MOD([Customer_ID], 12) + 1
```



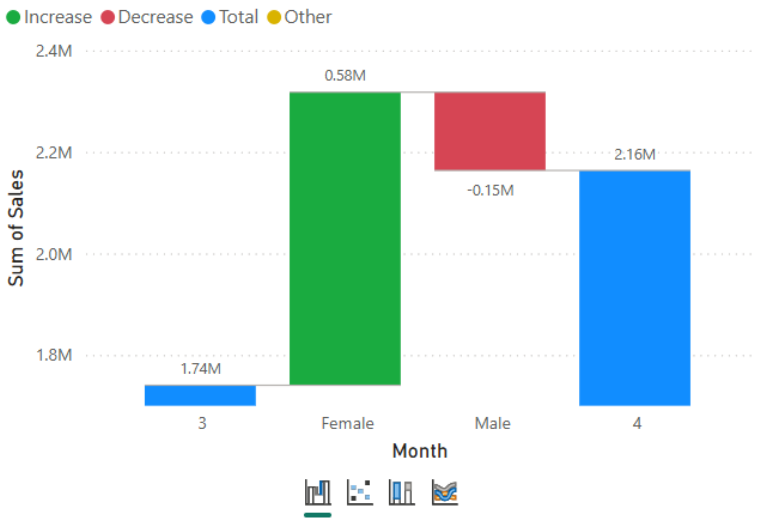
I have analysed the point which is pointed on the chart below.



Here's the analysis of the 24.32% increase in Sum of Sales between 3 and 4

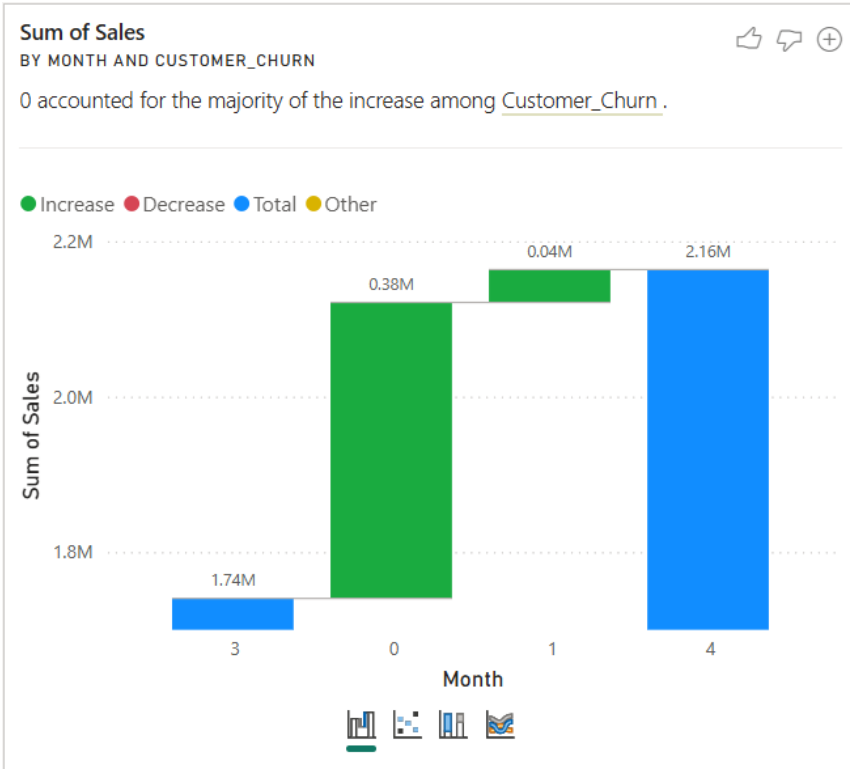
Sum of Sales
BY MONTH AND GENDER

'Female' accounted for the majority of the increase among Gender , offsetting the decrease of 'Male'.



Sum of Sales
BY MONTH AND SEASONALITY

'High' accounted for the majority of the increase among Seasonality , offsetting the decrease of 'Low'. The relative contribution made by 'Low' changed the most.



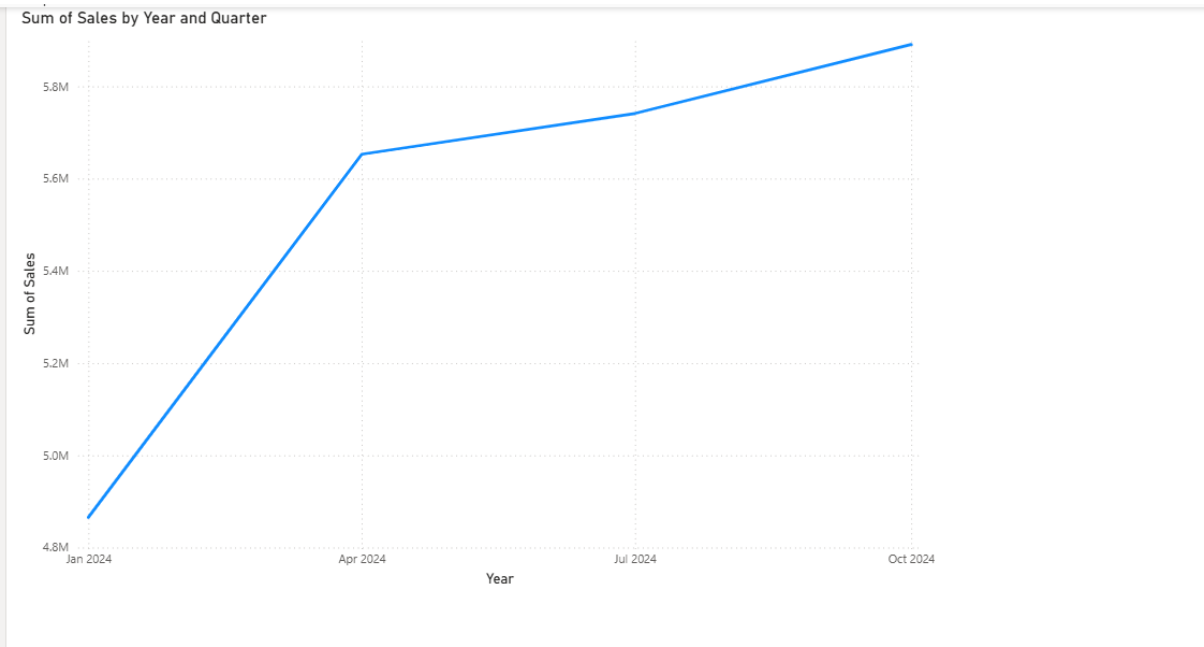
Summary:

- Gender: The increase was driven mainly by Female customers (≈ +0.58M), partially offset by a decrease among Male customers (≈ −0.15M).
- Seasonality: High season contributed the most to the growth (≈ +0.37M), Medium also added (≈ +0.22M), while Low declined (≈ −0.16M).
- Customer Churn: The gain came largely from non-churned customers (0) (≈ +0.38M); churned (1) added a small amount (≈ +0.04M).

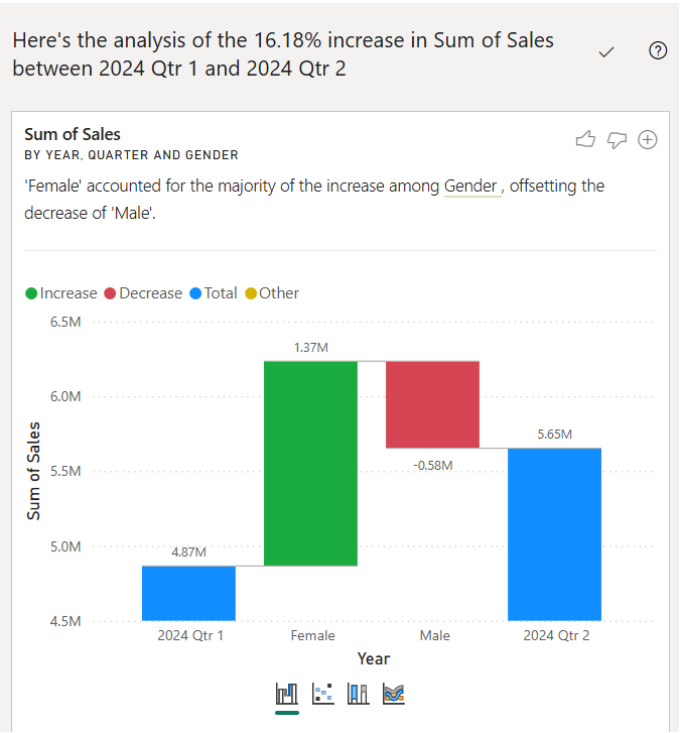
I have added another column by writing this:

MonthDate = DATE(2024, 'cleaned_data'[Month], 1) to make a real date not just 1-12.

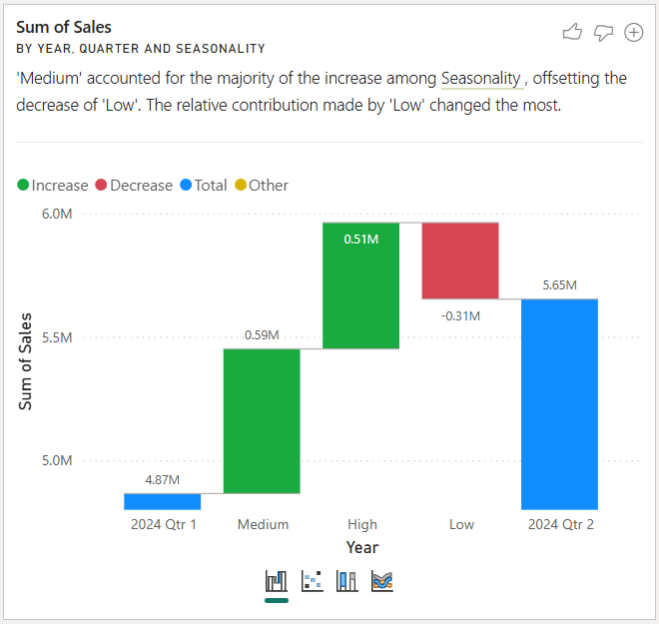
I have created line chart as below.

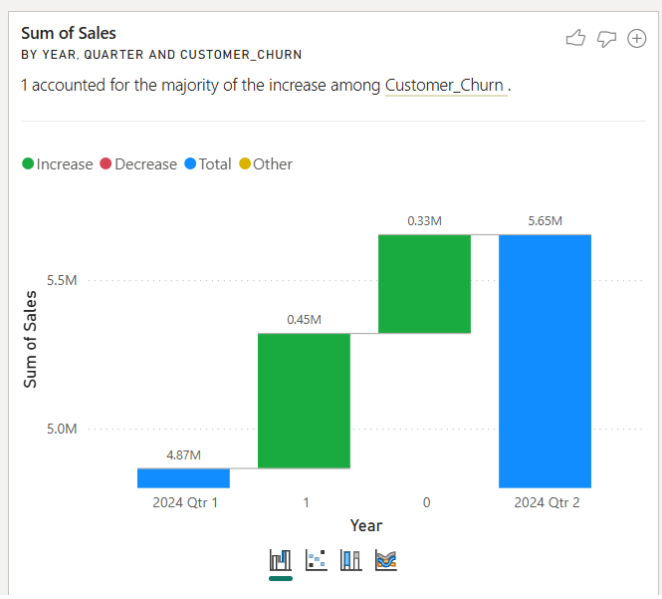


I am analysing April 2024.



Here's the analysis of the 16.18% increase in Sum of Sales between 2024 Qtr 1 and 2024 Qtr 2





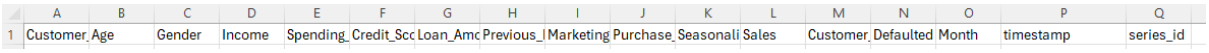
Summary:

- Sales rose +16.18% QoQ from 2024 Q1: ~4.87M to 2024 Q2: ~5.65M.
- Female +~1.37M increase offset by Male −~0.58M (net +~0.79M).
- Seasonality driver: Medium +~0.59M, High +~0.51M, while Low −~0.31M.
- Churn driver: Customer_Churn = 1 +~0.45M and Customer_Churn = 0 +~0.33M both contributed to the rise.

3.Predictive model results.

Vertex AI

- I have added 2 columns to my dataset:
- timestamp and series



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Customer_	Age	Gender	Income	Spending_	Credit_Scc	Loan_Amc	Previous_I	Marketing	Purchase_	Seasonali	Sales	Customer_	Defaulted	Month	timestamp	series_id

- New file name: k_with_time_multi_series

4.Summary report outlining key findings and business recommendations.