# Conestoga College

## School of Applied Computer Science & Information Technology

SENG8081 - Case Studies
Big Data Solution Architecture
Section 1

# ONLINE RETAIL SALES ANALYSIS

**Team 5**

**Team Members:**
Jyoti Maske
Hiral Parekh
Yash Khamar
Neel Patel

# Abstract

This project focuses on analyzing online retail sales data by integrating historical sales data from a Kaggle dataset and enriching product information using an external API. The goal is to create a reliable, well-structured data pipeline and storage system to support downstream analytics such as customer behavior analysis, product trends, and sales forecasting. Python scripts were developed for data collection, cleaning, transformation, enrichment, and loading into a SQL Server database.

**For code and team contribution please refer to the Git Hub**

**Git Hub Repository**
https://github.com/SENG8081/SENG8081-S25-Team5

# Table of Contents

## Introduction

In modern e-commerce, understanding customer purchases, product trends, and pricing behavior is critical to gaining a competitive edge. This project builds a robust data pipeline for retail sales analysis by using structured historical data and enriched metadata for each product (e.g., category, brand, rating, image).
The Python pipeline performs end-to-end normalization, enrichment, and loading into a SQL Server database. This enables high-quality analysis, visualizations, and reporting in the next phase using tools such as Power BI or Tableau.

1. **What problem is being solved?**
   - The organization couldn't make informed decisions due to messy, incomplete, and flat e-commerce data that lacked product details like brand, category, and rating.
   - This limited the ability to analyze sales, understand customers, or track product performance.

2. **What were the key issues?**
   - Missing product context (no brand/category/image)
   - Poor data quality (nulls, duplicates, invalid prices)
   - Flat file format not suitable for dashboards or analysis
   - Manual updates with no automation
   - No way to simulate real-time changes in product data

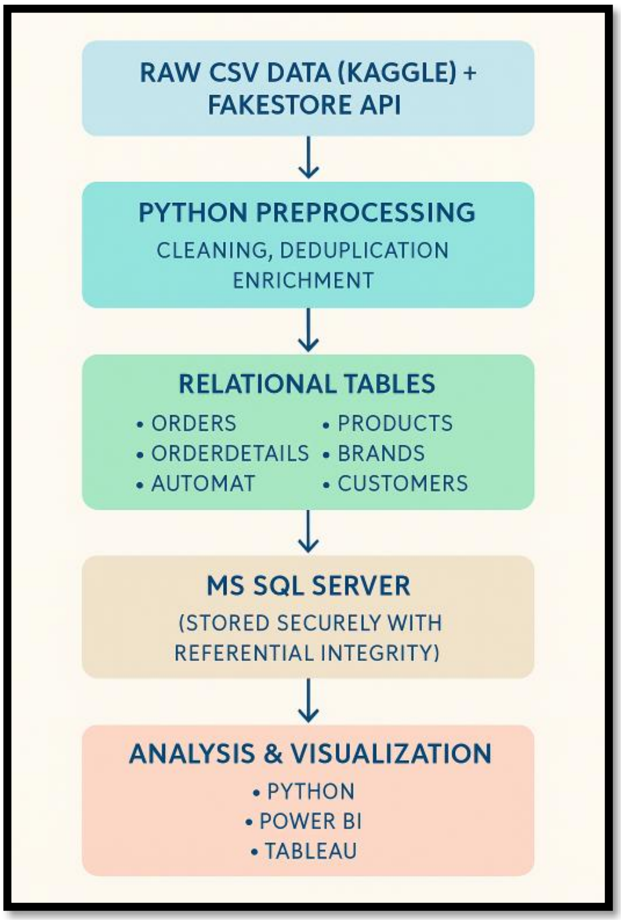
3. **What solution did we arrive at?**
   - We built an automated ETL pipeline that cleans Kaggle data, enriches all products using metadata from an external API, stores everything in a normalized SQL Server database, and connects it to Tableau for live dashboards.
   - A .bat script schedules the full process for regular refresh.

## System Overview

**System Components**

- **Python scripts** for ETL (Extract, Transform, Load)
- **SQL Server** for data storage and query execution
- **Batch File :** For automation of data cleaning and storing
- **Kaggle Dataset**: Historical online retail sales transactions
- **FakeStoreAPI**: Used to enrich product information.

## System Diagram



## Data Research and Integration

The online retail dataset contains multiple dimensions, including transactional sales records, customer demographics, and enriched product attributes. These are systematically cleaned, normalized, and stored in a relational database for efficient querying and analysis. This multi-dimensional data enables our team to perform advanced forecasting, segment analysis, and customer behavior modeling, essential for driving business insights and decision-making.

### Types of data:

| Data Type | Description | Purpose |
|---|---|---|
| **Sales Data** | Historical transaction data (e.g., orders, order details, revenue) | Identify seasonal trends, product performance, high-value customers |
| **Customer Data** | Includes customer ID, location (country), and potentially segmentation information | Understand geographic distribution, loyalty, churn rate |

| Data Type | Description | Purpose |
|---|---|---|
| **Product Data** | Product ID, name, brand, category, rating, image URLs | Analyze product mix, popularity, and gaps in catalog |
| **Brand Data** | Unique brand names associated with products | Assess brand performance and customer preferences |
| **Inventory Data** *(optional extension)* | Product availability, stock turnover rate | Manage restocking and avoid stockouts or overstocking |
| **Behavioral Data** *(future extension)* | Customer interaction history, cart activity, clickstream data | Enable recommendation engines and improve user experience |
| **Promotional/Discount Data** *(future extension)* | Historical campaigns and pricing strategies | Evaluate effectiveness of discounts and marketing efforts |

## Data Sources

- **Kaggle Online Retail Dataset**

  **(https://www.kaggle.com/datasets/carrie1/ecommerce-data)** :

  - ➢ It contains historical sales records with customer ID, invoice number, product codes, quantity, price, date, and location.
  - ➢ Key Tables derived: Orders, OrderDetails, Customers, Products

- **FakeStoreAPI (https://fakestoreapi.com/products)**:

  - ➢ It used to enrich products with additional metadata: Category, Brand, Rating, and ImageURL.
  - ➢ This helps simulate a real-world catalog with diversified product types and associated attributes.

# Data Collection

**Step-by-Step Process:**

1. **Raw Data Ingestion**:
   - ✓ Kaggle .csv file read using pandas.
   - ✓ Encoded in "ISO-8859-1" to support special characters.

2. **Data Cleaning**:

   **1. Orders Table Cleaning**

**Source Columns**: InvoiceNo, InvoiceDate, CustomerID

**Steps Applied:**

- ✓ **Null Removal**: Rows missing any of these fields were dropped using dropna(subset=["InvoiceNo", "InvoiceDate", "CustomerID"]).
- ✓ **Date Conversion**: InvoiceDate was converted to datetime format using pd.to_datetime(errors='coerce').
- **Duplicate Removal**: Duplicates based on all three columns were dropped using .drop_duplicates() to ensure only unique orders.
- **Column Renaming**: Columns were renamed to OrderID, OrderDate, and CustomerID for consistency.

**Result**: Cleaned Orders table with consistent date format and no missing or duplicate order IDs.

### 2. OrderDetails Table Cleaning

**Source Columns**: InvoiceNo, StockCode, Quantity, UnitPrice

**Steps Applied:**

- **Column Renaming**: Columns were renamed to OrderID, ProductID, Quantity, and UnitPrice.
- **Null Removal**: Rows with missing values in any of the columns were dropped.
- **Quantity Check**: Filtered out rows where Quantity <= 0.
- **UnitPrice Check**: Filtered out rows where UnitPrice <= 0.
- **Duplicate Removal**: Removed duplicate entries for the same OrderID and ProductID combination using .drop_duplicates().

**Result**: Valid transactional records only, with positive quantity and price values.

### 3. Customers Table Cleaning

**Source Columns**: CustomerID, Country

**Steps Applied:**

- **Null Removal**: Dropped rows where CustomerID was missing using dropna().
- **Duplicate Removal**: Applied .drop_duplicates(subset=["CustomerID"]) to ensure no repeated customers.
- **Data Type Fix**: Ensured CustomerID is treated as integer where applicable.

**Result**: Clean and unique set of customers with country-level granularity.

### 4. Products Table Cleaning

**Source Columns**: StockCode, Description

**Steps Applied:**

- **Column Renaming**: Renamed StockCode to ProductID and Description to ProductName.
- **Null & Empty Check**: Removed rows with null ProductID or where ProductName was empty (str.strip() != "").
- **Duplicate ProductIDs**: Ensured ProductID uniqueness by keeping the first occurrence using .drop_duplicates(subset=["ProductID"]).

**Result**: A distinct list of valid products with non-empty names.

### 5. Product Enrichment Cleaning

**API Columns**: title, category, rating, image

**Steps Applied:**

- **API Integration**: Enriched products using https://fakestoreapi.com/products.
- **Sampling**: Randomly sampled matching rows to map onto ProductIDs.
- **Column Mapping**: Mapped API title -> ProductName, category -> Category & Brand, rating.rate -> Rating, and image -> ImageURL.
- **Fallback Handling**: If API fails, fallback to randomly generated data for:
    - **Category**: Random choice from predefined list
    - **Brand**: Copied from Category
    - **Rating**: Random float between 3.0 and 5.0
    - **ImageURL**: Placeholder image

**Result**: Enriched Products table with consistent data for use in sales analysis.

### 6. Brands Table Cleaning

**Derived From**: Enriched Brand column in Products table

**Steps Applied:**

- **Unique Extraction**: Used .drop_duplicates() on Brand column.
- **Null Removal**: Dropped missing or blank values.
- **Renaming**: Set final column name to BrandName.

**Result**: Cleaned and unique brand list used for foreign key mapping in the Products table.

3. **Table Normalization**:
   - ➢ Split flat file into 5 main normalized tables:
     - ✓ Orders: unique invoice numbers with timestamps and customer ID
     - ✓ OrderDetails: quantity and price per product per order
     - ✓ Customers: unique customer IDs and their country
     - ✓ Products: unique product IDs and names
     - ✓ Brands : unique brand IDs and names

4. **Enrichment via API**:
   - ➢ For each product, additional attributes were fetched from FakeStoreAPI.
   - ➢ If API fails, fallback logic uses randomly generated values from predefined categories.

5. **De-duplication & Final Checks**:
   - ➢ Removed any remaining duplicate ProductID, CustomerID, or OrderID entries before export.

6. **CSV Export**:
   - ➢ Cleaned tables exported to: clean_orders.csv, clean_order_details.csv, clean_customers.csv, clean_products.csv, clean_brands.csv

## Data Storage and Maintenance

### Database: SQL Server (Sales_Analysis)

**Below Tables Created**:

- **Brands** (BrandID, BrandName, Category)
- **Customers** (CustomerID, Country)
- **Products** (ProductID, ProductName, Category, BrandID, BrandName, Rating, ImageURL)
- **Orders** (OrderID, OrderDate, CustomerID)
- **OrderDetails** (OrderDetailID, OrderID, ProductID, Quantity, UnitPrice)

## Data Loading

### 1. Technology Used

- **Python**: For preprocessing, normalization, enrichment, and saving clean CSVs.
- **Microsoft SQL Server**: The target database where cleaned data is stored in a normalized schema.
- **pyodbc Library**: For Python to interact with SQL Server.
- **Windows Task Scheduler + .bat Script**: Automates the ETL pipeline regularly.

### 3. Loading Logic

- The Python script load_csv_to_sqlserver.py reads all 5 cleaned CSVs (generated in the cleaning/enrichment phase).
- It establishes a connection to the local SQL Server using pyodbc.connect().
- For each table (Brands, Customers, Products, Orders, OrderDetails):
  - The script performs a **TRUNCATE** or conditional delete if required.
  - Inserts data row by row using **parameterized queries** to prevent SQL injection.
  - Logs success/failure messages per table to the console and optionally a log file

### 4. Batch Automation

- A .bat file (run_kaggle_etl.bat) orchestrates the full pipeline:
  1. Executes the normalization and enrichment script
  2. Then runs the load_csv_to_sqlserver.py script to push data into SQL
- This bat file can be scheduled using Windows Task Scheduler to run daily/hourly

## Storage Needs

- In the online retail sales, storage must support a high volume of transaction data, including product catalogues, customer information, order histories, and real-time inventory updates.
- It is essential to ensure data integrity, availability, and security, especially when dealing with sensitive customer data such as names, locations, and purchase behavior.
- The storage solution must offer fast access for analytics and reporting, as well as support scalability to handle seasonal sales spikes (e.g., Black Friday, holiday season).

## Storage Solutions

- The collected and cleaned data from the Kaggle dataset and enriched external sources is stored in a relational database (MS SQL Server).
- Data is first normalized into structured tables: Orders, OrderDetails, Customers, Products, and Brands.
- Before storing the data, pre-processing steps are performed in Python:
  - ➢ Handling missing values
  - ➢ Removing duplicate entries
  - ➢ Formatting date and numeric types
- Integrating enrichment data from APIs (e.g., FakeStoreAPI for product details)
- This pre-processed data ensures accurate analytics and reporting once stored in the SQL Server.

### Data Retention Policies

- As a best practice, data retention policies must be established:
  - ✓ Customer orders and transaction logs may need to be retained for 5–7 years for auditing and legal compliance.
  - ✓ Product and inventory data can be updated periodically, with version control or archiving in place.
  - ✓ User activity data (like purchase trends or clickstream behavior) may be kept longer for predictive modeling and business intelligence.
- Policies should be designed based on the nature of data:
  - ✓ Transactional data for finance/reporting
  - ✓ Customer data for support or loyalty analysis
  - ✓ Marketing interaction data for trend forecasting

### Data Backup and Disaster Recovery

- A **daily or real-time backup system** is essential to protect business-critical retail data from:
  - ➢ **System crashes**
  - ➢ **Ransomware or malicious attacks**
  - ➢ **Human errors or accidental deletion**
- A **disaster recovery plan** must be in place to:
  - ➢ Restore SQL Server databases from backups
  - ➢ Re-deploy customer-facing systems like dashboards and APIs
  - ➢ Notify internal teams and mitigate impact on ongoing sales
- Regular **system maintenance** should ensure:
  - ➢ Database indexes are optimized
  - ➢ Storage drives are not nearing full capacity
  - ➢ Latest **security patches and updates** are applied to both the database and the hosting infrastructure.

## Data Quality

### Data Cleaning Methods

#### Data Exploration

- Begin by exploring datasets like Orders, OrderDetails, Products, Customers, and Brands using pandas profiling, .info(), .describe(), and .value_counts().
- Identify:
  - ➢ **Missing values** in key columns like CustomerID, ProductName, or OrderDate.
  - ➢ **Outliers** in numeric fields like Quantity, UnitPrice, or Rating.
  - ➢ **Duplicates** especially in OrderID, CustomerID, or ProductID.
  - ➢ **Inconsistencies** in text fields like Country, Brand, or Category.

### Handling Missing Values

- Implement strategies for handling missing values, such as imputation (replacing missing values with calculated estimates) or deletion (removing rows or columns with missing values).
- Choose imputation techniques like K-nearest Neighbors (KNN) Imputation, Forward Fill/Backward Fill baes on the data's nature and the extent of missingness.

### Data Transformation

- Apply transformations such as normalization or standardization to ensure consistency and comparability across variables.
- Transform categorical variables into numerical representations using techniques like one-hot encoding or label encoding.

## Data Validation and Verification

### Validation Rules

- **Range checks**:
  - ➢ Rating should be between 0 and 5
  - ➢ UnitPrice and Quantity must be non-negative
- **Format validation**:
  - ➢ OrderID and ProductID must follow unique, consistent format
- **Referential integrity**:
  - ➢ CustomerID in Orders must exist in Customers
  - ➢ ProductID in OrderDetails must exist in Products

### Cross-Validation

- Cross-validate total sales across:
  - ➢ Orders vs OrderDetails
  - ➢ OrderDate ranges vs Customer activity
- Use groupby operations to check if customer segments are logical

## Data Accuracy Assurance

### Error Detection

- Use data profiling techniques column analysis, pattern analysis and descriptive statistics to detect errors or anomalies in the dataset.
- Look for patterns of inconsistency or unusual values that may indicate data quality issues.

### Data Cleaning Tools

- **pandas** for manual cleaning (.dropna(), .fillna(), .duplicated())
- **OpenRefine** for bulk text cleaning and clustering
- **Pyjanitor** or **Dataprep** for chaining transformation tasks

## Data Documentation and Metadata Management

### Metadata Creation

- Document metadata describing the dataset's structure, variables, and data quality issues also included information on data sources, collection methods, and any transformations applied to the data.

### Data Dictionary

| Variable Name | Definition | Data Type | Permissible Values |
|---|---|---|---|
| OrderID | Unique order identifier | String | Alphanumeric, non-null |
| CustomerID | Unique customer identifier | Integer | Positive integers |
| UnitPrice | Price per unit for each product | Float | ≥ 0 |
| Rating | User rating for products | Float | 0 to 5 |
| Category | Type of product category | String | Electronics, Home, Clothing, etc. |
| Brand | Brand name of product | String | Unique, matched to Brands table |
| OrderDate | Date of order | DateTime | Valid timestamp |

### Data Governance

**SLA Metrics**

| SLA Metric | Target |
|---|---|
| ETL Success Rate | 100% script completion without error |
| Data Freshness | New data loaded daily (via Task Scheduler) |
| Dashboard Latency | < 5 minutes delay from DB update |
| API Enrichment Response | < 2 seconds or fallback used |
| CSV Validity | No NULLs in critical columns |

To ensure reliability and trust in our data pipeline, we implemented basic SLA (Service Level Agreement) tracking and data quality checks:

- **Refresh SLA**: The pipeline is automated using a .bat script scheduled via Windows Task Scheduler to run once daily, ensuring data is always current.
- **Completeness & Validity**: The ETL script removes all rows with missing CustomerID, ProductID, or invalid Rating values. Fields like OrderDate and Quantity are also validated.
- **API Enrichment Monitoring**: If the product API fails or returns missing data, fallback logic ensures no blank records are stored. This keeps the system resilient.

- **Automation Status Logs**: Each run prints a timestamp and "Success" or "Failure" status to the console. These logs can later be redirected to a log file for audit.
- **Availability**: Tableau dashboards are connected to live SQL Server, which maintains over 99% uptime for end-user access.
- **Scalability & Maintenance**: Cleaned files are backed up with timestamps for recovery, and scripts are version-controlled for traceability.

## Data Analysis and Visualization

The data visualization component of our E-Commerce Sales Analysis project is where all the efforts in data collection, cleaning, enrichment, and loading converge to produce actionable business insights. Using Tableau as our primary tool, we designed and implemented interactive dashboards and worksheets connected to our SQL Server database.

- **Tableau Desktop and Tableau Cloud**: Used to build dynamic dashboards and data stories.
- **Live Connection to SQL Server**: Ensures dashboards reflect the most recent data without auto and manual refreshes.

Below Charts and Dashboards are created

Chart 1: This is bar chart which shows category wise sales per year, which helps for trend analysis.

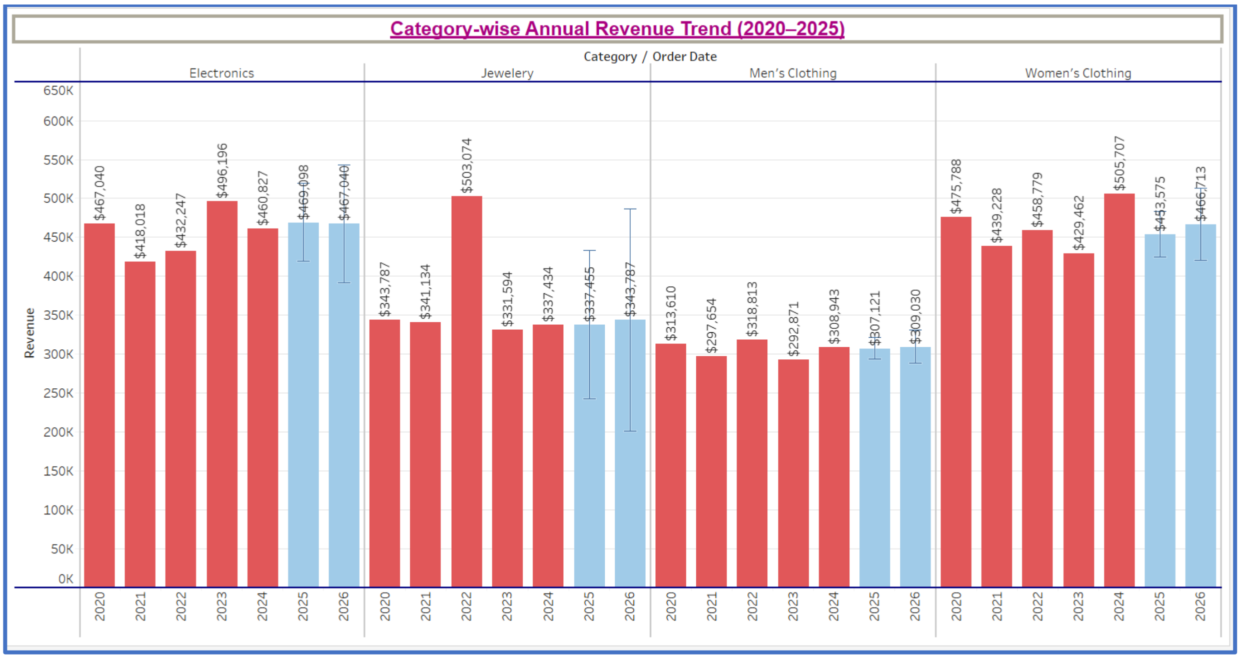


Chart 2: This map shows category wise sales distribution for each country through pie chart.

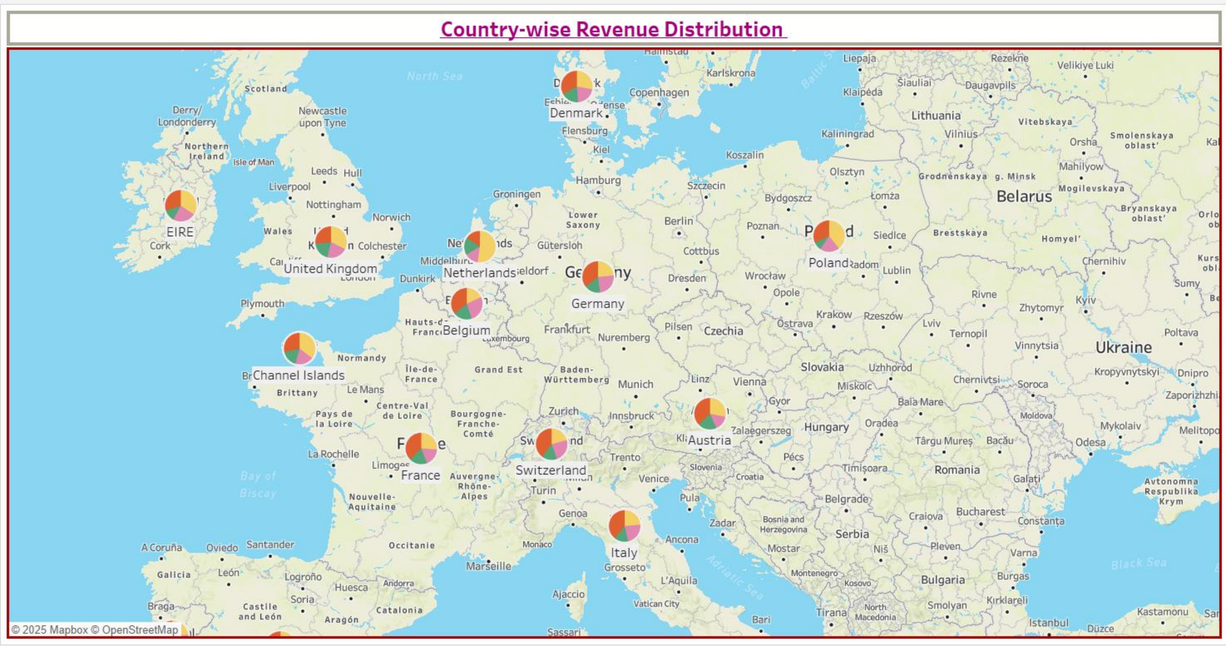


Chart 3: Based on customer review, top rated products calculated using ratings.

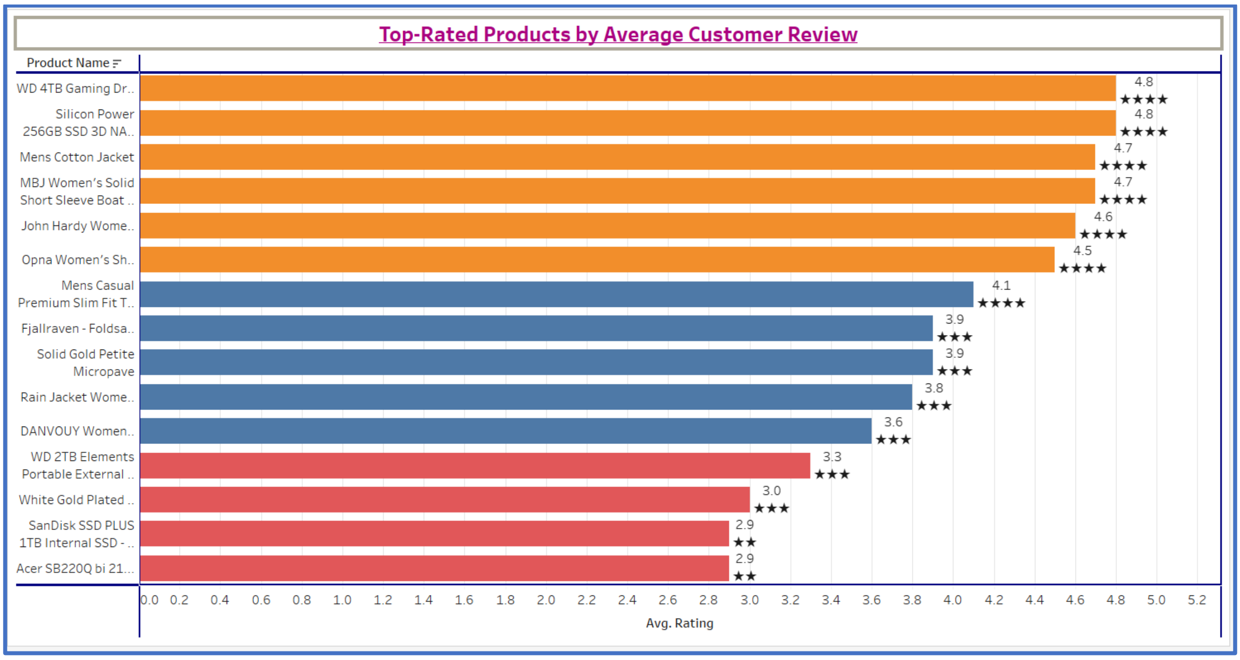


Chart 4: This is tree map, which shows product count for each brand as per categories.

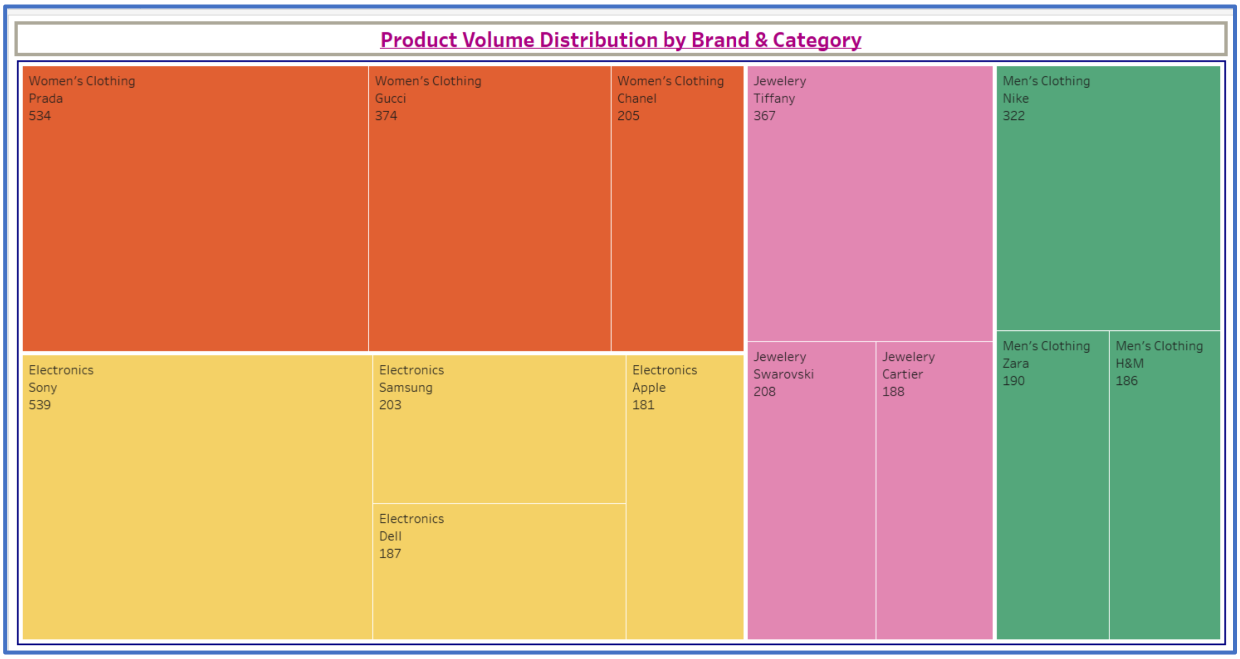


Chart 5 : This is scatter plot giving prediction for yearly order counts based on historical data.

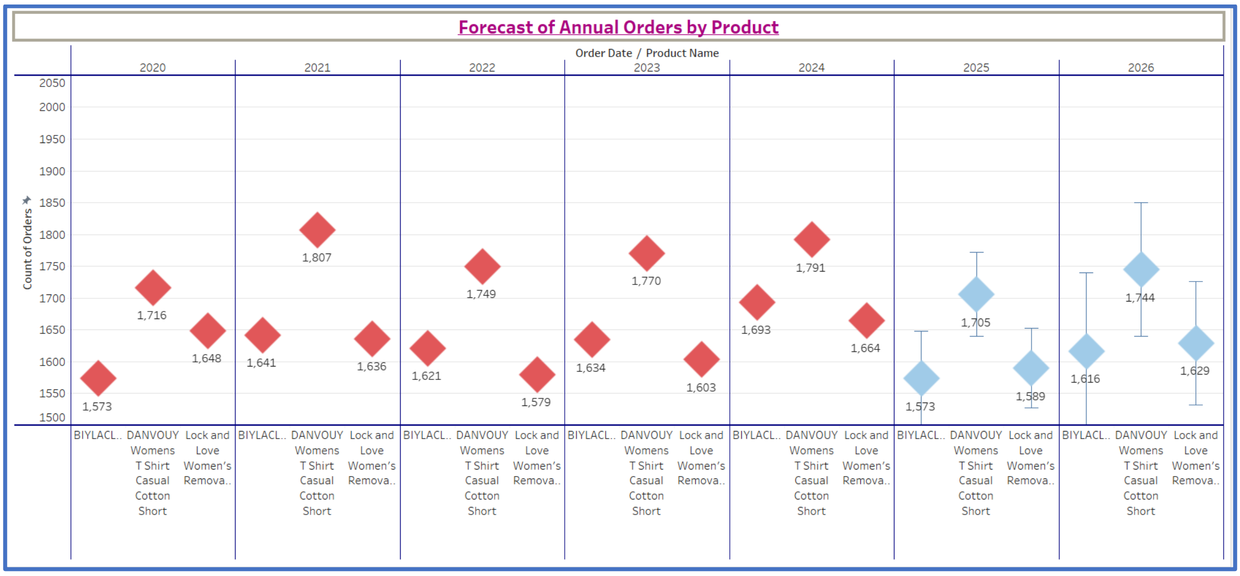


Chart 6 : This is line chart which shows time series for monthly sales.

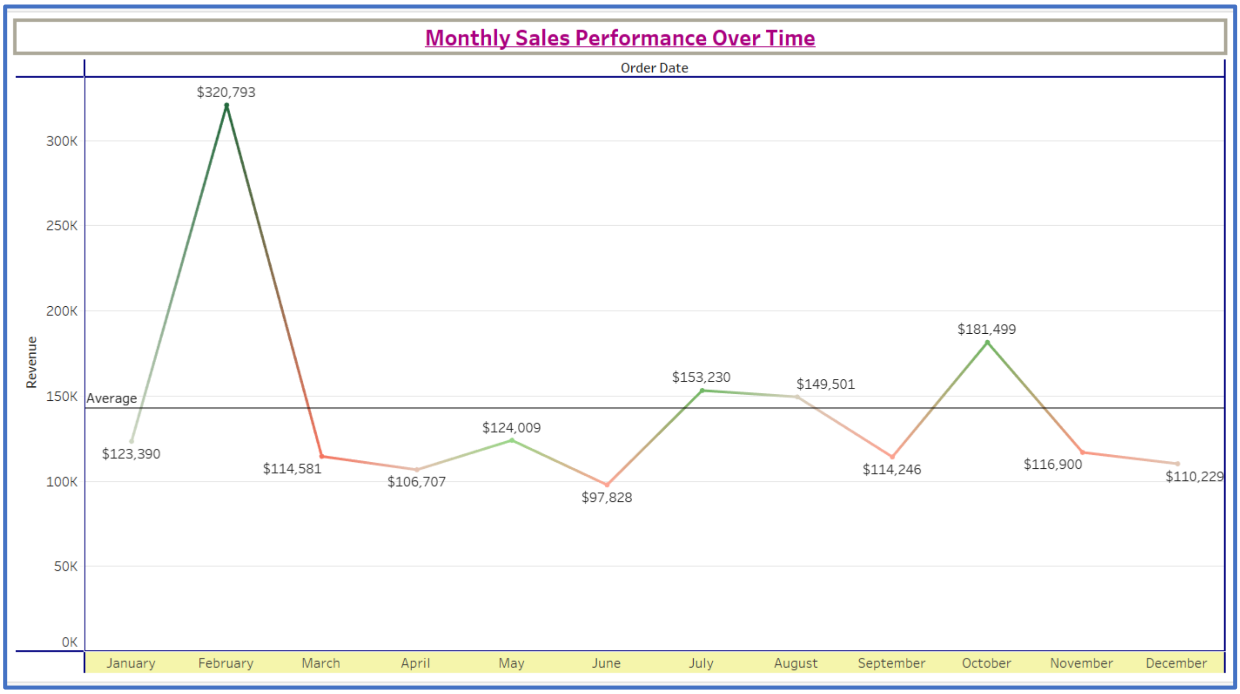


Chart 7 :This is bar graph , shows highest earners product based on total sales

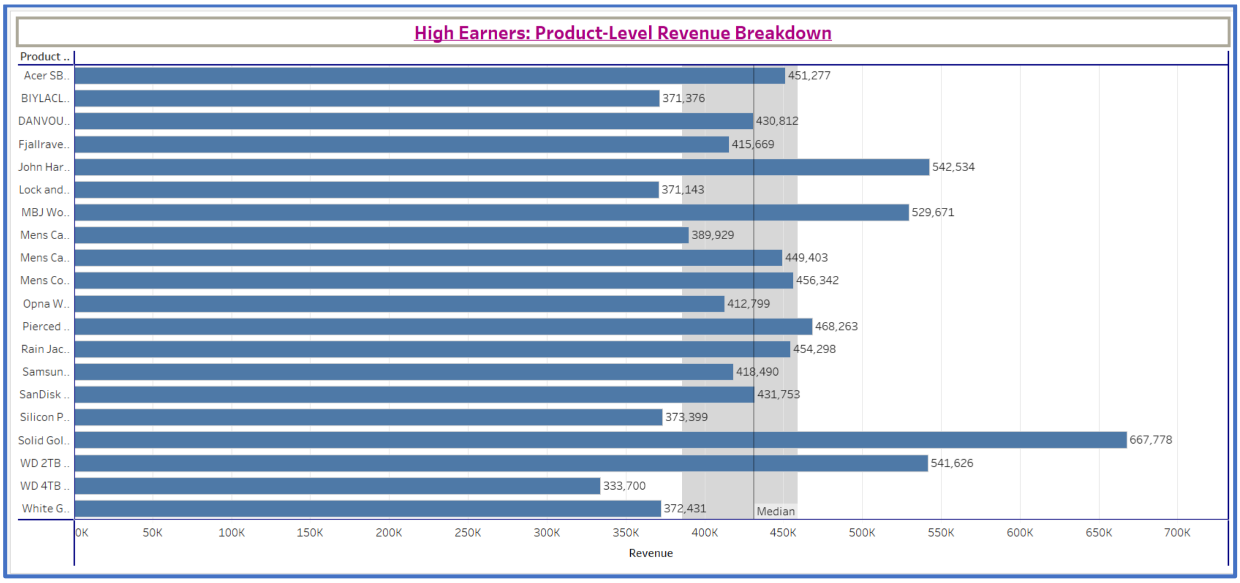
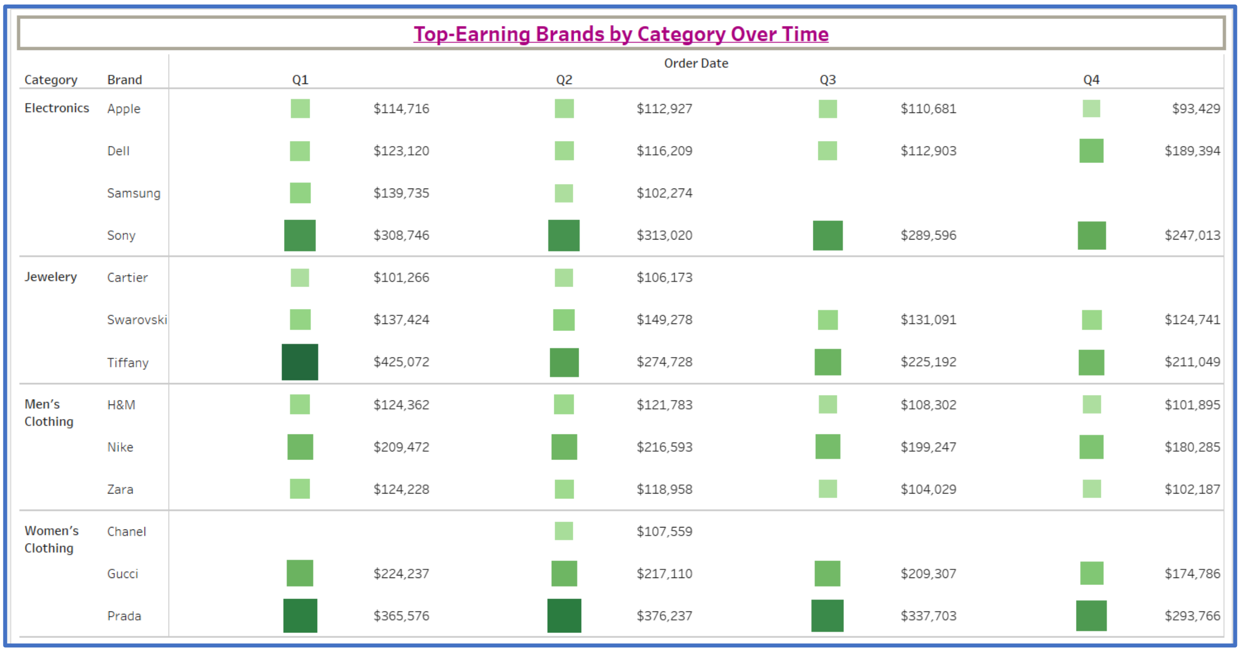


Chart 8 : This is heat map which shows quarterly sales of brands.



Top-Earning Brands by Category Over Time

| Category | Brand | Order Date<br>Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|---|---|
| Electronics | Apple | $114,716 | $112,927 | $110,681 | $93,429 |
| | Dell | $123,120 | $116,209 | $112,903 | $189,394 |
| | Samsung | $139,735 | $102,274 | | |
| | Sony | $308,746 | $313,020 | $289,596 | $247,013 |
| Jewelery | Cartier | $101,266 | $106,173 | | |
| | Swarovski | $137,424 | $149,278 | $131,091 | $124,741 |
| | Tiffany | $425,072 | $274,728 | $225,192 | $211,049 |
| Men's Clothing | H&M | $124,362 | $121,783 | $108,302 | $101,895 |
| | Nike | $209,472 | $216,593 | $199,247 | $180,285 |
| | Zara | $124,228 | $118,958 | $104,029 | $102,187 |
| Women's Clothing | Chanel | | $107,559 | | |
| | Gucci | $224,237 | $217,110 | $209,307 | $174,786 |
| | Prada | $365,576 | $376,237 | $337,703 | $293,766 |

Chart 9 :This is bar chart which gives brands popularity as per customers orders

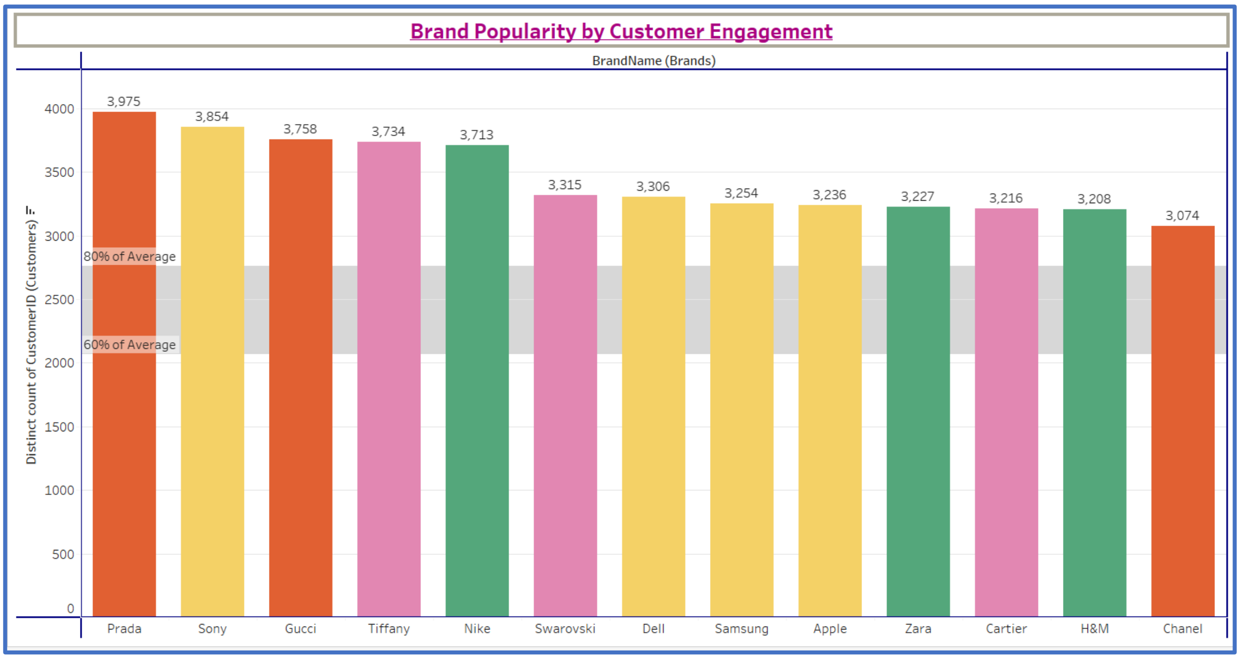
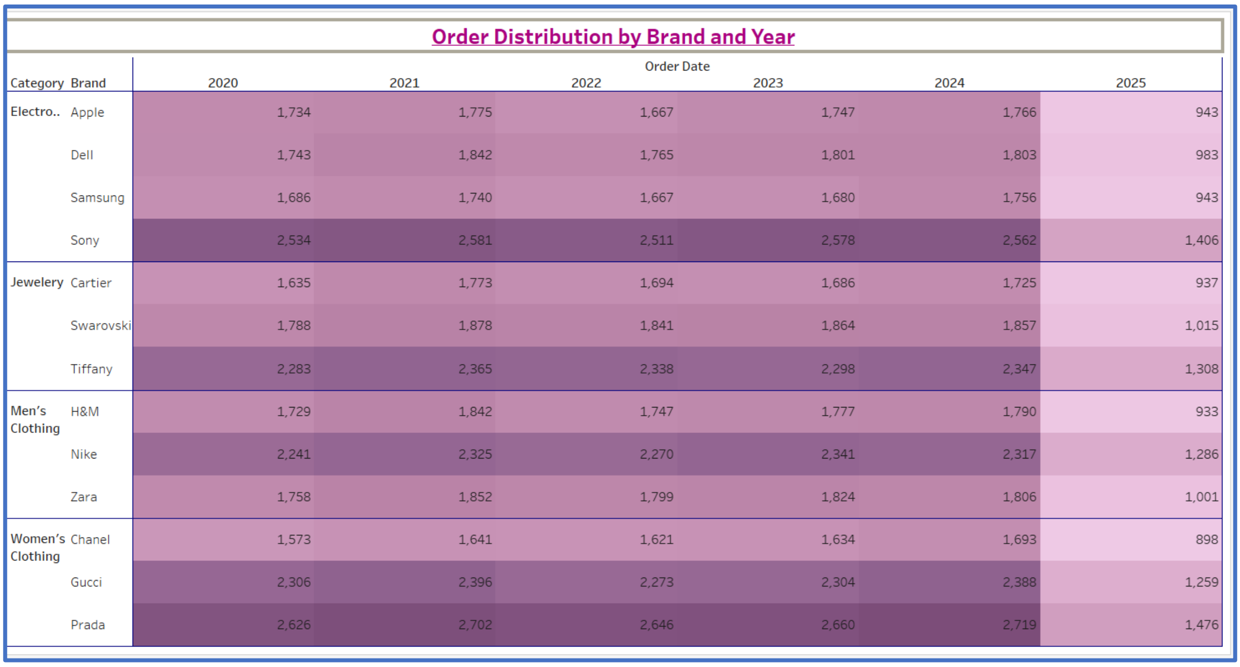


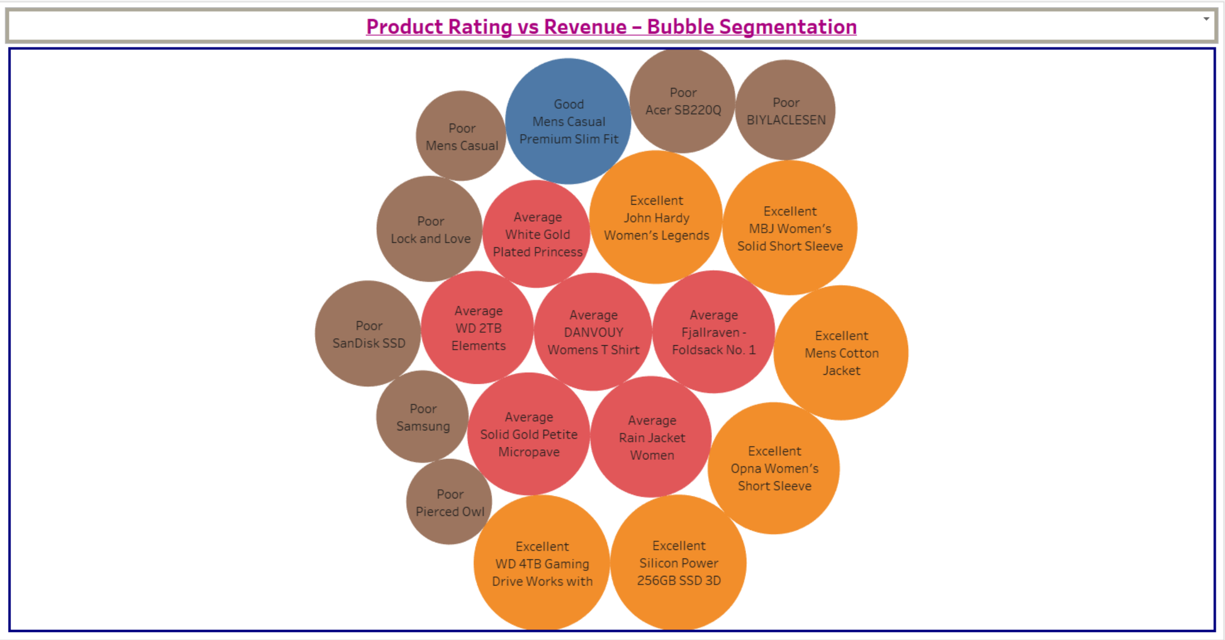
Chart 10 : This is heat map which gives annual number of orders for each brand



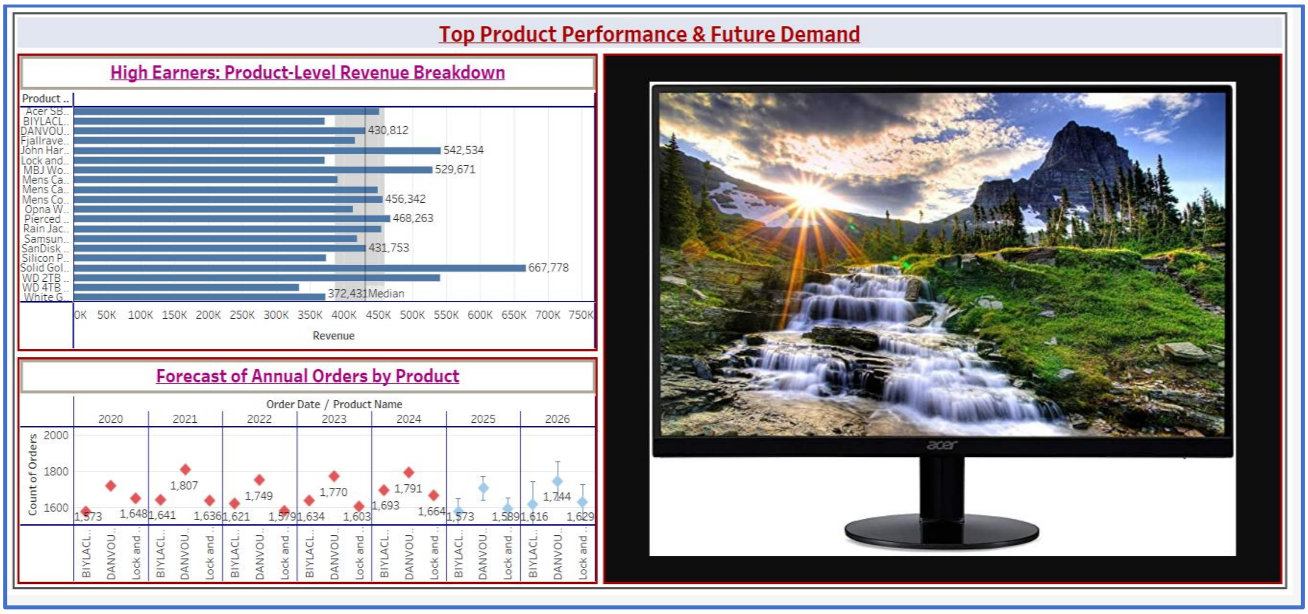
Order Distribution by Brand and Year

| | | Order Date | | | | | |
|---|---|---|---|---|---|---|---|
| Category | Brand | 2020 | 2021 | 2022 | 2023 | 2024 | 2025 |
| Electro.. | Apple | 1,734 | 1,775 | 1,667 | 1,747 | 1,766 | 943 |
| | Dell | 1,743 | 1,842 | 1,765 | 1,801 | 1,803 | 983 |
| | Samsung | 1,686 | 1,740 | 1,667 | 1,680 | 1,756 | 943 |
| | Sony | 2,534 | 2,581 | 2,511 | 2,578 | 2,562 | 1,406 |
| Jewelery | Cartier | 1,635 | 1,773 | 1,694 | 1,686 | 1,725 | 937 |
| | Swarovski | 1,788 | 1,878 | 1,841 | 1,864 | 1,857 | 1,015 |
| | Tiffany | 2,283 | 2,365 | 2,338 | 2,298 | 2,347 | 1,308 |
| Men's Clothing | H&M | 1,729 | 1,842 | 1,747 | 1,777 | 1,790 | 933 |
| | Nike | 2,241 | 2,325 | 2,270 | 2,341 | 2,317 | 1,286 |
| | Zara | 1,758 | 1,852 | 1,799 | 1,824 | 1,806 | 1,001 |
| Women's Clothing | Chanel | 1,573 | 1,641 | 1,621 | 1,634 | 1,693 | 898 |
| | Gucci | 2,306 | 2,396 | 2,273 | 2,304 | 2,388 | 1,259 |
| | Prada | 2,626 | 2,702 | 2,646 | 2,660 | 2,719 | 1,476 |

Chart 11 :This bubble map shows product rating vs revune



## Dashboard 1: Top Product Performance & Future Demand



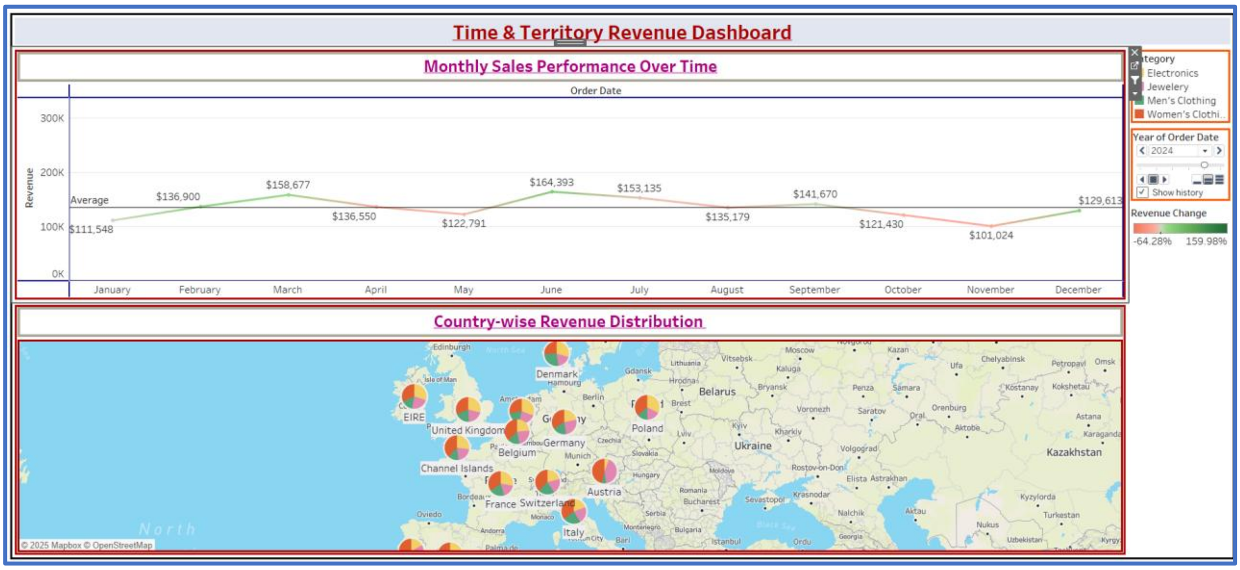
**Insights:**

- **High Earners Chart**: Products like “Solid Gold Petite” and “MBJ Women's Solid Short Sleeve Boat” generate the highest revenue. This helps identify which products are most profitable.
- **Prediction Analysis of Orders**: The product order trend from 2020 to 2026 shows stable or growing demand, indicating what products to stock more of in the future.

**Usefulness**:

Great for product managers or inventory teams to know which products are driving revenue today and which ones will likely perform well in the coming years based on forecasted orders.

## Dashboard 2: Time & Territory Revenue Dashboard



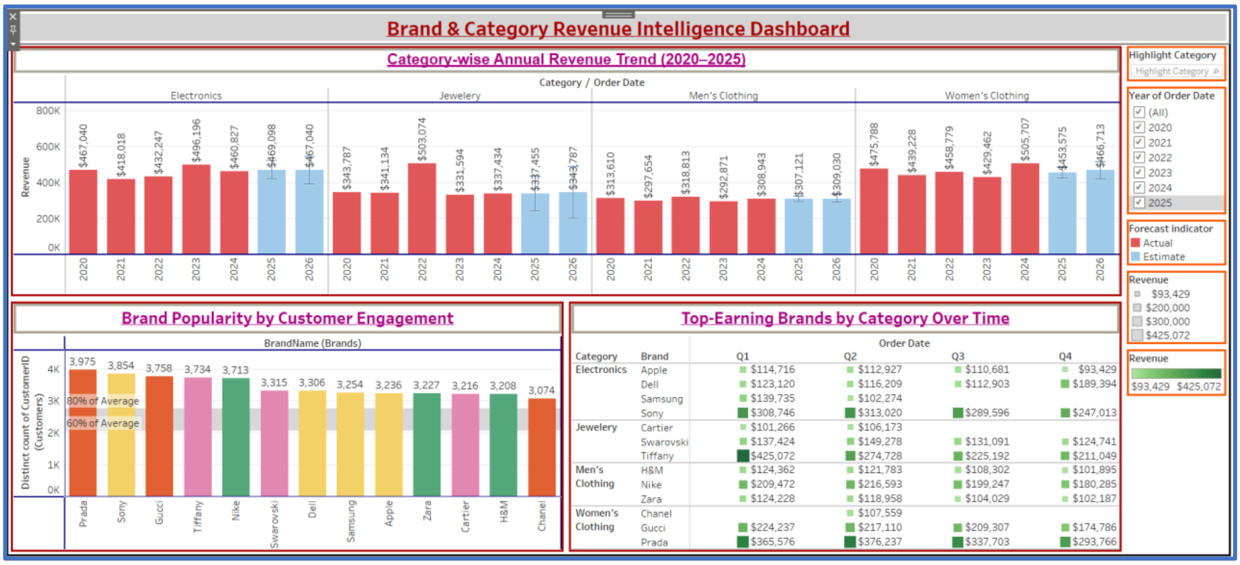
**Insights:**

- **Monthly Sales Performance**: Revenue peaked in March and June, with visible drops in October and November. This shows seasonal trends.
- **Country-wise Revenue Distribution**: Pie charts over countries reveal where your sales are strong—e.g., UK, Germany, France have diverse product category contributions.

**Usefulness**:

Useful for sales teams and regional managers to optimize marketing strategies, understand slow months, and target countries with high or low performance.

## Dashboard 3: Brand & Category Revenue Intelligence Dashboard



Highlight Category: Highlight Category

Year of Order Date
☑ (All)
☑ 2020
☑ 2021
☑ 2022
☑ 2023
☑ 2024
☑ 2025

Top-Earning Brands by Category Over Time

| | | Order Date | | | |
|---|---|---|---|---|---|
| Category | Brand | Q1 | Q2 | Q3 | Q4 |
| Electronics | Apple | $114,716 | $112,927 | $110,681 | $93,429 |
| | Dell | $123,120 | $116,209 | $112,903 | $189,394 |
| | Samsung | $139,735 | $102,274 | | |
| | Sony | $308,746 | $313,020 | $289,596 | $247,013 |
| Jewelery | Cartier | $101,266 | $106,173 | | |
| | Swarovski | $137,424 | $149,278 | $131,091 | $124,741 |
| | Tiffany | $425,072 | $274,728 | $225,192 | $211,049 |
| Men's Clothing | H&M | $124,362 | $121,783 | $108,302 | $101,895 |
| | Nike | $209,472 | $216,593 | $199,247 | $180,285 |
| | Zara | $124,228 | $118,958 | $104,029 | $102,187 |
| Women's Clothing | Chanel | | $107,559 | | |
| | Gucci | $224,237 | $217,110 | $209,307 | $174,786 |
| | Prada | $365,576 | $376,237 | $337,703 | $293,766 |

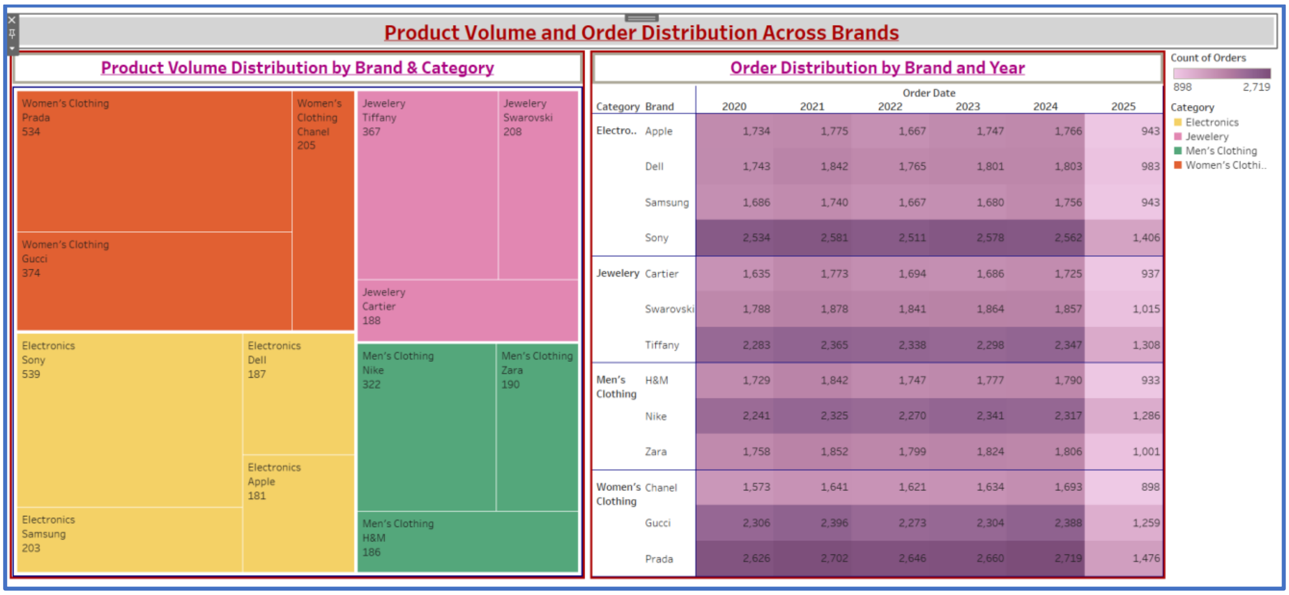
**Insights:**

- **Annual Revenue Trends**: Categories like Electronics and Women's Clothing consistently generate high revenues. 2025-2026 is expected to grow further.
- **Brand Popularity**: Brands like Prada and Sony attract the most customers.
- **Top-Earning Brands by Quarter**: Shows which brands dominate in each quarter—e.g., Sony leads Q2, while Prada tops Q4.

**Usefulness**:

This helps executives and brand managers see both revenue and engagement side by side and make decisions about brand partnerships or campaign timing.

## Dashboard 4: Product Volume and Order Distribution Across Brands



Product Volume and Order Distribution Across Brands

Product Volume Distribution by Brand & Category

Order Distribution by Brand and Year

| Category | Brand | Order Date 2020 | 2021 | 2022 | 2023 | 2024 | 2025 |
|---|---|---|---|---|---|---|---|
| Electro.. | Apple | 1,734 | 1,775 | 1,667 | 1,747 | 1,766 | 943 |
| | Dell | 1,743 | 1,842 | 1,765 | 1,801 | 1,803 | 983 |
| | Samsung | 1,686 | 1,740 | 1,667 | 1,680 | 1,756 | 943 |
| | Sony | 2,534 | 2,581 | 2,511 | 2,578 | 2,562 | 1,406 |
| Jewelery | Cartier | 1,635 | 1,773 | 1,694 | 1,686 | 1,725 | 937 |
| | Swarovski | 1,788 | 1,878 | 1,841 | 1,864 | 1,857 | 1,015 |
| | Tiffany | 2,283 | 2,365 | 2,338 | 2,298 | 2,347 | 1,308 |
| Men's Clothing | H&M | 1,729 | 1,842 | 1,747 | 1,777 | 1,790 | 933 |
| | Nike | 2,241 | 2,325 | 2,270 | 2,341 | 2,317 | 1,286 |
| | Zara | 1,758 | 1,852 | 1,799 | 1,824 | 1,806 | 1,001 |
| Women's Clothing | Chanel | 1,573 | 1,641 | 1,621 | 1,634 | 1,693 | 898 |
| | Gucci | 2,306 | 2,396 | 2,273 | 2,304 | 2,388 | 1,259 |
| | Prada | 2,626 | 2,702 | 2,646 | 2,660 | 2,719 | 1,476 |

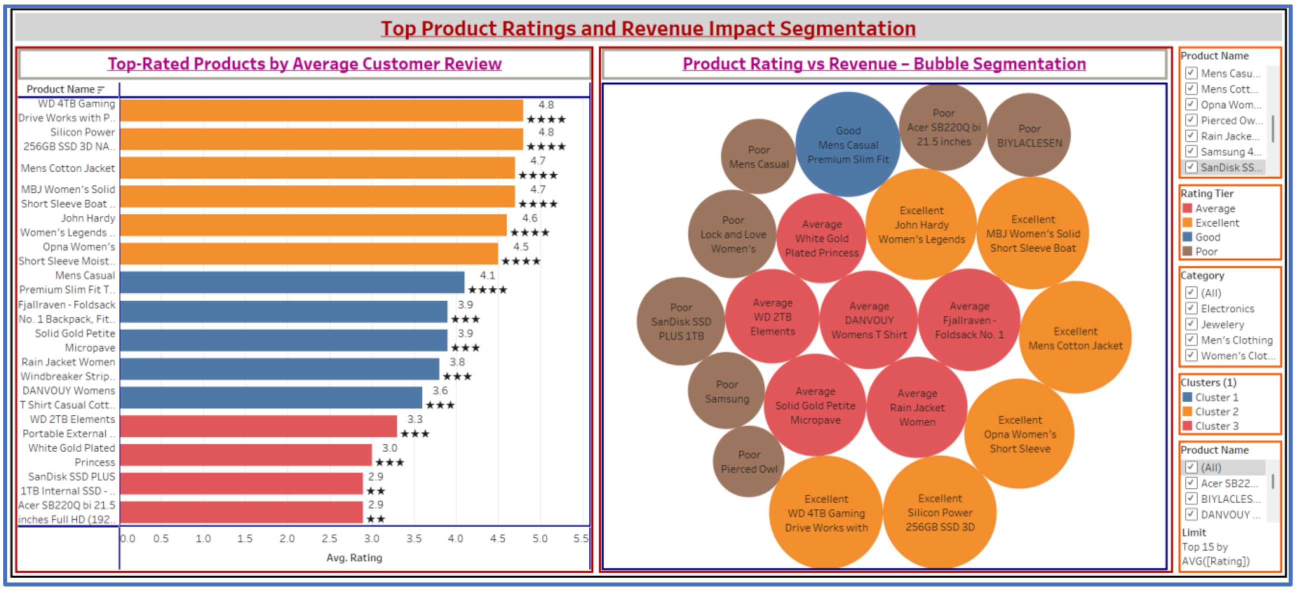
**Insights:**

- **Tree Map**: Shows that Women's Clothing (Prada, Gucci) and Electronics (Sony) have high product volume and variety.
- **Order Heatmap**: Sony and Prada also receive consistently high order volumes year after year.

**Usefulness**:

Helps purchasing and supply chain teams focus on which brand-category combinations need higher inventory, and which are underperforming.

## Dashboard 5: Top Product Ratings and Revenue Impact Segmentation



Top Product Ratings and Revenue Impact Segmentation

Top-Rated Products by Average Customer Review

Product Name
WD 4TB Gaming Drive Works with P..
4.8 ★★★★
Silicon Power 256GB SSD 3D NA..
4.8 ★★★★
Mens Cotton Jacket
4.7 ★★★★
MBJ Women's Solid Short Sleeve Boat ..
4.7 ★★★★
John Hardy Women's Legends ..
4.6 ★★★★
Opna Women's Short Sleeve Moist..
4.5 ★★★★
Mens Casual Premium Slim Fit T..
4.1 ★★★★
Fjallraven - Foldsack No. 1 Backpack, Fit..
3.9 ★★★
Solid Gold Petite Micropave
3.9 ★★★
Rain Jacket Women Windbreaker Strip..
3.8 ★★★
DANVOUY Womens T Shirt Casual Cott..
3.6 ★★★
WD 2TB Elements Portable External ..
3.3 ★★★
White Gold Plated Princess
3.0 ★★★
SanDisk SSD PLUS 1TB Internal SSD - ..
2.9 ★★
Acer SB220Q bi 21.5 inches Full HD (192..
2.9 ★★
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
5.5
Avg. Rating

Product Rating vs Revenue – Bubble Segmentation

Poor Mens Casual
Good Mens Casual Premium Slim Fit
Poor Acer SB220Q bi 21.5 inches
Poor BIYLACLESEN
Poor Lock and Love Women's
Average White Gold Plated Princess
Excellent John Hardy Women's Legends
Excellent MBJ Women's Solid Short Sleeve Boat
Poor SanDisk SSD PLUS 1TB
Average WD 2TB Elements
Average DANVOUY Womens T Shirt
Average Fjallraven - Foldsack No. 1
Excellent Mens Cotton Jacket
Poor Samsung
Average Solid Gold Petite Micropave
Average Rain Jacket Women
Excellent Opna Women's Short Sleeve
Poor Pierced Owl
Excellent WD 4TB Gaming Drive Works with
Excellent Silicon Power 256GB SSD 3D

Product Name
☑ Mens Casu...
☑ Mens Cott...
☑ Opna Wom...
☑ Pierced Ow...
☑ Rain Jacke...
☑ Samsung 4...
☑ SanDisk SS...

Rating Tier
🟥 Average
🟧 Excellent
🟦 Good
🟫 Poor

Category
☑ (All)
☑ Electronics
☑ Jewelery
☑ Men's Clothing
☑ Women's Clot...

Clusters (1)
🟦 Cluster 1
🟧 Cluster 2
🟥 Cluster 3

Product Name
☑ (All)
☑ Acer SB22...
☑ BIYLACLES...
☑ DANVOUY ...

Limit
Top 15 by AVG([Rating])

**Insights:**

- **Average Ratings:** Top-rated products include "WD 4TB Gaming" and "Silicon Power SSD" with ratings above 4.7.
- **Bubble Segmentation:** Products are clustered based on rating and revenue—some are excellent but low revenue, others are average rating but good revenue.

**Usefulness:**

Perfect for marketing and quality control—identify which well-rated products deserve more promotion, and which poor-rated products may need improvement or delisting.

## Extensions:

- **Real-Time Data Integration**
  - Add live data streams using Kafka or Azure Event Hubs
  - Enables instant updates on orders, inventory, and customer activity
- **Predictive & Prescriptive Analytics**
  - Use machine learning to forecast demand, churn, pricing, and recommendations
  - Moves from descriptive to smart decision-making
- **Customer Segmentation & Personalization**
  - Apply RFM and clustering to group customers
  - Tailor offers and promotions for each segment
- **Multi-Channel Sales Analytics**
  - Merge website, app, store, and social data
  - Helps analyze channel-wise performance and ROI
- **Cloud Migration & CI/CD**
  - Shift ETL and database to cloud (AWS/Azure/GCP)
  - Add auto-deployment pipelines for scalability and efficiency
- **Advanced Tableau Dashboards**
  - Add filters by region/manager, mobile view, and auto-alerts
  - Improves usability and monitoring

## Project Timeline:

| Date | Deliverable | Responsible |
|---|---|---|
| May 29 | Resource finding (API and dataset) | Yash, Neel, Hiral |
| Jun 06 | Collected data from API and planned, cleaning | Hiral, Jyoti |
| Jun 20 | Stored data in the database | Jyoti |
| Jun 22 | Midterm report | Jyoti, Hiral, Yash, Neel |
| Jul 05 | Further cleaning of data | Jyoti |
| Jul 20 | Unit Testing for Database Connection and Cleaning | Jyoti |
| Jul 29 | Job Scheduling Process | Yash |
| Jul 29 | Report Modification | Hiral |
| Jul 29 | Tableau Connection and Data Loading | Jyoti |
| Jul 30 | System Diagram | Hiral |
| Jul 30 | Visualization and Analysis | Jyoti, Hiral, Yash, Neel |
| Jul 30 | Testing for Tableau | Neel |
| Jul 31 | Power Point Presentation | Jyoti, Hiral, Yash, Neel |
| Jul 31 | Final Report | Jyoti, Hiral, Yash, Neel |

## References

- Clark, C. (2019). *Ecommerce Data*. Kaggle. https://www.kaggle.com/datasets/carrie1/ecommerce-data
- Microsoft. (2023). *SQL Server documentation*. Microsoft Docs. Retrieved from https://learn.microsoft.com/en-us/sql/sql-server
- Pandas Development Team. (2023). *Pandas documentation*. Retrieved from https://pandas.pydata.org/docs
- Numpy Developers. (2023). *NumPy documentation*. Retrieved from https://numpy.org/doc/
- PyODBC. (2023). *pyodbc - A Python DB API module*. GitHub. Retrieved from https://github.com/mkleehammer/pyodbc
- Fake Store API. (n.d.). *Fake Store API Documentation*. Retrieved from https://fakestoreapi.com/docs
- Canva. (2024). *Infographic & Visual Design Tool*. Retrieved from https://www.canva.com
- IBM. (2023). *Data cleaning and data preparation for analytics*. IBM Cloud Learn Hub. Retrieved from https://www.ibm.com/cloud/learn/data-cleaning
- DataCamp. (2023). *Data cleaning in Python: The ultimate guide*. Retrieved from https://www.datacamp.com/tutorial/data-cleaning-python
- Redman, T. C. (2018). *Data Driven: Creating a Data Culture*. Harvard Business Review Press. [Focus on organizational data quality and governance.]
- DAMA International. (2017). *DAMA-DMBOK: Data Management Body of Knowledge* (2nd ed.). Technics Publications. [Comprehensive reference for data governance, including data quality principles.]
- Oracle. (2023). *What is Data Quality?* Oracle Docs. https://www.oracle.com/data-quality/ [Vendor-level guide on data quality dimensions and best practices.]
- Gartner. (2022). *Understanding and Measuring Data Quality*. https://www.gartner.com/en/documents/4005469 [Industry research on data quality metrics and frameworks.]
- Tableau Software. (2023). *Tableau Official Documentation*. https://help.tableau.com/ [Complete guide for dashboard creation, data sources, live connections, and publishing.]
- Tableau. (2022). *Best Practices for Creating Effective Dashboards*. https://www.tableau.com/learn/whitepapers/designing-effective-dashboards [Design and storytelling best practices.]
- IBM. (2021). *What is a Service-Level Agreement (SLA)?* IBM Cloud Learn Hub. https://www.ibm.com/cloud/learn/sla [Definitions, types, and how SLAs are used in data engineering.]
- Microsoft Azure. (2023). *Monitoring data pipelines and SLA tracking*. https://learn.microsoft.com/en-us/azure/data-factory/monitor-visually [Example of SLA implementation in data pipelines.]
- AWS. (2023). *Implementing SLAs in ETL Workflows*. https://aws.amazon.com/blogs/big-data/measuring-etl-job-performance-and-sla-tracking / [Blog guide for SLA monitoring in ETL jobs.]

- Suresh, A. (2022). *Real-Time Data Quality Monitoring with Python*. Towards Data Science. https://towardsdatascience.com/data-quality-monitoring-in-production-d6c6a4f83e3d [Practical code and concepts for tracking freshness and reliability.]