Data Wrangling Report

Introduction

Data wrangling is the process of cleaning, structuring, and enriching raw data to make it suitable for analysis. This report outlines the steps taken to preprocess the dataset used in this project.

Dataset Overview

- **Dataset Name:** Supermarket Sales Data
- **File Type:** CSV
- **Number of Rows:** 1000 (Checked using df.shape)
- **Number of Columns:** 12 (Checked using df.shape)
- **Columns:** ['Invoice ID', 'Branch', 'City', 'Customer type', 'Gender', 'Product line', 'Unit price', 'Quantity', 'Tax 5%', 'Total', 'Date', 'Time', 'Payment', 'cogs', 'gross margin percentage', 'gross income', 'Rating']
- **Target Variable:** Total Sales (Derived from 'Total' column)

Data Inspection

1. Previewing the Data

- Used df.head() and df.tail() to view the first and last 5 rows.
- Used df.shape to check the number of rows and columns.
- Used df.info() to get an overview of the data types and missing values.
- Used df.describe() to check statistical summaries of numerical columns.

2. Checking for Missing Values

- Used df.isnull().sum() to identify missing values in each column.
- Visualized missing data using a heatmap (sns.heatmap(df.isnull(), cbar=False)).
- Found missing values in the 'Rating' column.
- Applied the **Forward Fill (ffill) method** to handle missing values in 'Rating'.

3. Checking for Duplicates

- Used df.duplicated().sum() to count duplicate rows.
- Removed duplicate entries using df.drop_duplicates(inplace=True) if any were found.

4. Data Type Conversion

- Converted the Date column to **datetime format** using pd.to_datetime(df["Date"]).
- Extracted Month and Day from the date column.

- Changed categorical columns ('Customer type', 'Gender', 'Product line', 'Payment') to **category type** to optimize memory usage.
- Converted 'Time' column into hourly bins to analyze time-based trends.

Data Cleaning & Transformation

1. Handling Outliers

- Used df.describe() to check for extreme values.
- Used boxplots (sns.boxplot()) to visualize potential outliers.
- Detected outliers in 'Unit price' and 'Total'.
- Used IQR method to handle outliers where necessary.

2. Standardizing Column Names

- Ensured consistency in column names by converting them to lowercase and replacing spaces with underscores (df.columns = df.columns.str.lower().str.replace(" ", "_")).

3. Encoding Categorical Variables

- Converted categorical variables into numerical form using **label encoding** for binary variables.
- Used **one-hot encoding** for multi-category columns.

4. Creating New Features

- Derived **Total Revenue per Transaction** by summing 'cogs' and 'Tax 5%'.
- Created **Time of Day Categories** (Morning, Afternoon, Evening) from the 'Time' column.

Conclusion

After applying data wrangling techniques, the dataset is now clean and structured, making it ready for further analysis. The next steps involve data exploration and deriving business insights.

Prepared by: Bassant Yasser Mahmoud
Date: [2/13/2024]
Source: Derived from the dataset using Pandas, NumPy, Matplotlib, and Seaborn.