## horizontal line



Sentiment-Based Sales Optimization

18.12.2024

**Leveraging Customer Feedback to Enhance Satisfaction and Drive Sales**

VALARMATHI GANESSIN

ENVISION VIRTUE

Online Data Analyst Internship

December 2024 Project

**Sentiment-Based Sales Optimization**

### **Table of Contents**

1. Introduction
2. Project Goal and Objectives
3. Data Description
   * Data Source
   * Attributes used
   * Dependencies
   * Environment
4. Data Cleaning
   * Essentials
   * Handling Missing Values By Statistics method
   * Quantile statistics
   * Descriptive statistics
   * Most frequent values
   * Histogram
   * Correlations
   * Missing values
5. Data Modelling
   * Calculated Measures
   * Calculated Columns
   * Bin Variables
   * Dimension Tables and Fact Table
   * Snowflake Model
6. Methodology and Exploratory Data Analysis (EDA)
   * Home Page Summary
   * Customer Satisfaction Analysis
   * Sentiment Analysis
   * Demographic Analysis
   * Detailed Review Analysis
   * Review Text Analysis
   * Key Insights Drill through Page
7. Logistic Regression Regression Model
   * Justification on Choosing LR
   * Comparison with other Algorithm
   * Project Attributes and Correlations
   * Model Performance Metrics
   * Step By Step Explanation
   * Conclusion
   * Python Script
   * Output with Explanation
8. Logistic Regression Regression Model
9. Results and Findings
10. Visualizations
11. Future Analysis
12. Conclusion
13. Thank You

# 1.Introduction

In today's competitive e-commerce landscape, understanding customer sentiment and preferences is paramount for driving business success. Customer reviews provide a wealth of information that can be leveraged to gain insights into consumer satisfaction and behavior. The project titled **"Sentiment-Based Sales Optimization"** aims to analyze customer reviews of women's clothing from an e-commerce platform, utilizing data science and natural language processing techniques to extract actionable insights.

### **Project Goals and Objectives**

### **Goals and Analysis Steps with NLP and Models**

**Trend Identification**

* **Analysis Steps**:
  + Calculate average ratings and recommendations.
  + Analyze age distribution over reviews.
* **Models/Techniques**:
  + Descriptive Statistics
  + Time Series Analysis

**Pattern Discovery**

* **Analysis Steps**:
  + Perform keyword analysis to identify frequent terms in positive and negative reviews.
  + Use clustering to group similar reviews.
* **Models/Techniques**:
  + NLP: Term Frequency-Inverse Document Frequency (TF-IDF)
  + Model Algorithm: K-Means Clustering

**Demographic Insights**

* **Analysis Steps**:
  + Segment data by age group and analyze average ratings and sentiments.
  + Visualize satisfaction differences across demographics.
* **Models/Techniques**:
  + Descriptive Statistics
  + Visualization Tools (e.g., Matplotlib, Seaborn)

**Sentiment Analysis**

* **Analysis Steps**:
  + Preprocess text data (remove noise, stop words).
  + Apply sentiment analysis techniques.
  + Visualize sentiment distribution.
* **Models/Techniques**:
  + NLP: Natural Language Toolkit (NLTK) for preprocessing
  + Model Algorithm: VADER (Valence Aware Dictionary and sEntiment Reasoner) for sentiment analysis
  + Visualization Tools (e.g., Matplotlib, Seaborn)

**Feature Engineering**

* **Analysis Steps**:
  + Create features like word count, sentiment scores, and TF-IDF scores.
  + Normalize numeric features.
* **Models/Techniques**:
  + NLP: CountVectorizer or TF-IDF Vectorizer
  + Min-Max Scaler or Standard Scaler for normalization

**Predictive Modeling**

* **Analysis Steps**:
  + Split the dataset into training and testing subsets.
  + Select and train models to predict Recommended IND.
  + Evaluate models using performance metrics.
* **Models/Techniques**:
  + Logistic Regression
  + Decision Trees
  + Random Forest
  + Performance Metrics (accuracy, precision, recall, F1 score)

**Reporting Insights**

* **Analysis Steps**:
  + Summarize key findings from EDA, sentiment analysis, and predictive modeling.
  + Create visualizations to support insights.
  + Provide actionable recommendations.
* **Models/Techniques**:
  + Visualization Tools (e.g., Matplotlib, Seaborn)
  + Reporting Tools (e.g., Jupyter Notebook, PowerPoint)

**Ethical Considerations**

* **Analysis Steps**:
  + Ensure personal data is anonymized and handled securely.
  + Maintain transparency and document steps.
  + Respect data privacy regulations and ethical standards.
* **Models/Techniques**:
  + Data anonymization techniques
  + Compliance with GDPR and other data privacy regulations

**Objectives**

The primary objective of the project **"Sentiment-Based Sales Optimization"** is to analyze customer reviews of women's clothing from an e-commerce platform using data science and natural language processing (NLP) techniques. The goal is to extract actionable insights that can enhance customer satisfaction and drive sales. Specifically, the project aims to:

* Identify trends in customer satisfaction based on ratings and recommendations.
* Discover patterns in positive and negative feedback.
* Understand the relationship between customer demographics and product ratings.
* Categorize and analyze review sentiments.
* Develop predictive models to forecast product recommendations.

**Purpose**

The purpose of this project is to leverage the rich data found in customer reviews to inform and improve e-commerce strategies. Customer reviews are a valuable source of feedback, providing direct insights into customer preferences, experiences, and satisfaction levels. By systematically analyzing these reviews, the project seeks to:

* Enhance product offerings and customer experience by addressing common concerns and preferences identified in the reviews.
* Optimize marketing and sales strategies based on demographic insights and sentiment analysis.
* Improve decision-making through predictive models that forecast customer recommendations.
* Provide clear, actionable recommendations for stakeholders to drive growth and customer satisfaction.

Without this analysis, critical insights from customer feedback would remain untapped, potentially leading to missed opportunities for improvement and growth. This project aims to bridge that gap, offering a data-driven approach to understanding and enhancing the customer experience in the retail sector.

# 3.Data Description

**Data Source:**

**Data Format:** Womens\_Clothing CSV files

**Attributes used**

**Personal Information:**

* + Age

**Feature Variables:**

* + Clothing ID
  + Title
  + Review Text
  + Rating
  + Recommended IND
  + Positive Feedback Count
  + Division Name
  + Department Name
  + Class Name

**Target Variable:**

* + Recommended IND

**Dependencies used**

### **Python Libraries**

1. **Data Manipulation:**
   * **pandas: Data manipulation and analysis.**
   * **numpy: Numerical computations.**
2. **Data Visualization:**
   * **matplotlib: Plotting and visualization.**
   * **seaborn: Statistical data visualization**
   * **jinja2 library, which is required for pandas dataframe Styling.**
3. **Text Preprocessing and NLP:**
   * **nltk: Natural Language Toolkit for text processing.**
   * **spacy: Industrial-strength NLP in Python.**
   * **textblob: Simplified text processing, including sentiment analysis.**
4. **Machine Learning:**
   * **scikit-learn: Machine learning algorithms and tools.**
   * **xgboost: Gradient boosting framework (optional, if needed for modeling).**
5. **Model Evaluation:**
   * **scipy: Scientific computing tools.**
   * **statsmodels: Statistical models for validation and testing.**

### **Additional Tools**

* **jupyter: Jupyter Notebook for interactive coding and visualizations.**
* **pip: Python package manager to install dependencies.**

**Environment**

**Tools and Environment**

**Benefits of Using Google Colab for This Project**

1. **Accessibility and Collaboration:** As a cloud-based platform, Google Colab allows us to access our work from any location and collaborate with team members in real-time.
2. **Ease of Setup:** Google Colab comes with pre-installed libraries, making it easy to set up our development environment without worrying about local machine constraints.
3. **Integration with Google Drive:** Seamless integration with Google Drive simplified the process of loading and saving datasets, ensuring smooth workflow management.
4. **Free GPU and TPU Support:** Access to free GPUs and TPUs enables faster processing and model training, which is particularly beneficial for computationally intensive tasks.
5. **Interactive Coding and Visualization:** The Jupyter notebook interface supports interactive coding and visualization, allowing us to document our process and visualize results effectively.
6. **Reproducibility:** Colab ensures consistent environments across sessions, facilitating reproducibility of our results and making it easy to share our notebooks with others.

By leveraging these benefits, we were able to efficiently conduct our analysis, develop models, and document our findings.

# Utilizing PyCharm for "Sentiment-Based Sales Optimization" Using Machine Learning and NLP

PyCharm, a powerful Integrated Development Environment (IDE) for Python, significantly enhances development efficiency in projects like the **"Sentiment-Based Sales Optimization" Using Machine Learning and NLP.** It provides a comprehensive suite of tools that streamline coding, testing, and debugging processes. PyCharm's intelligent code editor features code completion, real-time error detection, and code refactoring capabilities, which help developers write cleaner and more maintainable code. Additionally, its robust debugging tools allow for effective troubleshooting and performance optimization. The integrated version control system supports seamless collaboration, while the built-in support for various frameworks, including those used for machine learning and NLP, facilitates smoother project execution. By leveraging PyCharm's extensive functionality, developers can focus more on innovation and problem-solving, ultimately accelerating the development cycle and enhancing overall productivity.

* + **Python 3.8**
  + **Power BI Desktop**
  + **MS Excel**
  + **PyCharm**
  + **Google COLAB**

# 4.Data Cleaning By Pandas Profiling

# **Benefits of Using Pandas for Data Cleaning:**

1. **Handling Missing Values:** Pandas provides functions like .dropna() and .fillna() to easily manage missing data.
2. **Removing Duplicates:** With .drop\_duplicates(), you can quickly identify and remove duplicate records.
3. **Data Transformation:** Pandas allows for seamless data type conversions, reshaping datasets, and merging multiple datasets.
4. **Data Exploration:** Built-in functions for descriptive statistics and data visualization help in understanding the data better.
5. **Efficiency:** Pandas is optimized for performance, making it suitable for large datasets.

# Rationale for Choosing Pandas Profiling in Data Preprocessing

Pandas profiling helped **"Sentiment-Based Sales Optimization" Using Machine Learning and NLP** by providing a comprehensive report on the Womens\_Clothing.csv file, offering insights into data quality, identifying missing values, and summarizing statistical properties, which streamlined the data cleaning process.

# 1.Essentials: (Data type,unique values and missing values by python script).

**Essentials Pandas Profiling.py**

#Python Script for finding the missing values

import pandas as pd

# Load the CSV file into a DataFrame

file\_path = "Womens\_Clothing.csv"

df = pd.read\_csv(file\_path)

# Find the data types of each column

data\_types = df.dtypes

print("Data Types:\n", data\_types)

# Find the number of unique values in each column

unique\_values = df.nunique()

print("\nUnique Values:\n", unique\_values)

# Find the number of missing values in each column

missing\_values = df.isnull().sum()

print("\nMissing Values:\n", missing\_values)

**OUTPUT**

**Data Types:**

Unnamed: 0 int64

Clothing ID int64

Age int64

Title object

Review Text object

Rating int64

Recommended IND int64

Positive Feedback Count int64

Division Name object

Department Name object

Class Name object

dtype: object

**Unique Values:**

Unnamed: 0 23486

Clothing ID 1206

Age 77

Title 13993

Review Text 22634

Rating 5

Recommended IND 2

Positive Feedback Count 82

Division Name 3

Department Name 6

Class Name 20

dtype: int64

**Missing Values:**

Unnamed: 0 0

Clothing ID 0

Age 0

Title 3810

Review Text 845

Rating 0

Recommended IND 0

Positive Feedback Count 0

Division Name 14

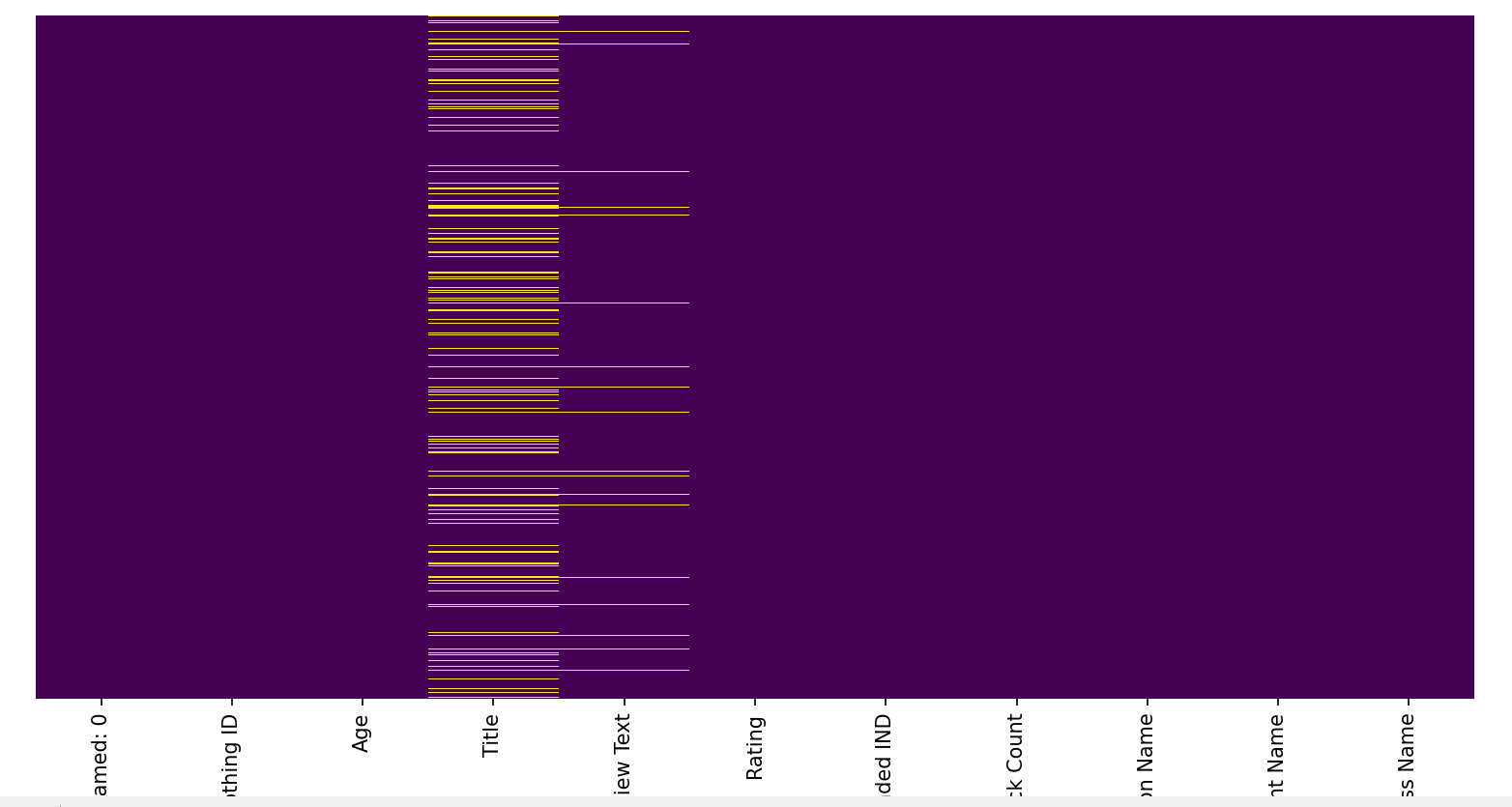
Department Name 14

Class Name 14

dtype: int64

Process finished with exit code 0

**Heat Map**



**Handling the missing values by Statistics.**

**Missing\_Values\_By\_Statistics.py**

import pandas as pd

import numpy as np

**# Load the dataset**

df = pd.read\_csv('womensclothing.csv')

**# Check for missing values**

print("Missing values before handling:\n", df.isnull().sum())

**# Handle missing values with statistical methods**

df['Age'] = df['Age'].fillna(df['Age'].mean())

for col in ['Division Name', 'Department Name', 'Class Name']:

df[col] = df[col].fillna(df[col].mode()[0])

df['Title'] = df['Title'].fillna(df['Title'].mode()[0])

df['Review Text'] = df['Review Text'].fillna(df['Review Text'].mode()[0])

**# Verify the changes**

print("Missing values after handling:\n", df.isnull().sum())

**# Save the cleaned dataset (optional)**

df.to\_csv('womensclothing\_cleaned.csv', index=False)

**Output**

**Missing values before handling:**

Unnamed: 0 0

Clothing ID 0

Age 0

Title 3810

Review Text 845

Rating 0

Recommended IND 0

Positive Feedback Count 0

Division Name 14

Department Name 14

Class Name 14

dtype: int64

**Missing values after handling:**

Unnamed: 0 0

Clothing ID 0

Age 0

Title 0

Review Text 0

Rating 0

Recommended IND 0

Positive Feedback Count 0

Division Name 0

Department Name 0

Class Name 0

dtype: int64

Process finished with exit code 0

# Justification for Using Statistical Methods

# Preserves Data Integrity:

# Numerical Columns (e.g., Age): Using the mean or median to fill missing values helps maintain the overall distribution of the data. This method avoids introducing bias that might occur if we arbitrarily remove or fill missing values. The mean is useful when data is symmetrically distributed, while the median is better for skewed distributions to avoid the influence of outliers.

# Ensures Consistency:

# Categorical Columns (e.g., Division Name, Department Name, Class Name): Filling missing values with the mode (the most frequent value) ensures that the categorical data remains consistent and logical. It helps in maintaining the integrity of the categorical variables without introducing new or unrealistic categories.

# Maintains Usability of Text Data:

# Text Columns (e.g., Title, Review Text): Filling missing text entries with the most frequent value (mode) or a placeholder ensures that the text fields remain usable for text preprocessing and NLP tasks. It prevents errors in text analysis algorithms that may arise from missing data.

# Avoids Data Loss:

# Complete Case Analysis: Instead of removing rows with missing data, statistical imputation allows us to retain as much data as possible. This is particularly important when the dataset is not very large, as it maximizes the amount of information available for analysis.

# Minimizes Bias:

# Systematic Approach: Statistical methods provide a systematic approach to handle missing data, reducing the risk of subjective decisions that could introduce bias. This ensures that the data imputation process is reproducible and transparent.

# Facilitates Advanced Analysis:

# Data Modeling and Machine Learning: Statistical imputation creates a more complete and consistent dataset, which is crucial for effective machine learning model training and validation. Missing values handled through statistics result in models that are more robust and reliable.

### **Why Not Simply Drop Missing Values?**

# Loss of Valuable Information: Dropping rows or columns with missing values can lead to significant loss of data, which might contain valuable information for analysis.

# Impact on Model Accuracy: Removing too much data can reduce the accuracy and generalizability of models. Statistical imputation helps to preserve the dataset’s integrity and ensures that models have sufficient data to learn from.

### Specific Choices in the Dataset

# Numerical Imputation: Using the mean for the Age column ensures that we maintain the central tendency of age data without skewing the distribution.

# Categorical Imputation: Using the mode for Division Name, Department Name, and Class Name ensures that we fill missing values with the most logical and frequent categories, maintaining categorical data consistency.

# Text Imputation: Using the mode or a placeholder for Title and Review Text maintains the usability of text data for NLP tasks.

# By choosing these statistical methods, we ensure a comprehensive, unbiased, and consistent approach to handling missing values, which is essential for accurate and reliable analysis in the project.

# *2.*Quantile statistics( minimum value, Q1, median, Q3, maximum, range, interquartile range)

**Quantile\_Analysis.py**

**#Quantile Analysis**

import pandas as pd

**# Load the CSV file into a DataFrame**

file\_path = "Womens\_Clothing.csv"

df = pd.read\_csv(file\_path)

**# Function to calculate and display quantile statistics for all columns**

def calculate\_quantile\_statistics(df):

statistics = {}

for column in df.columns:

if pd.api.types.is\_numeric\_dtype(df[column]):

min\_value = df[column].min()

Q1 = df[column].quantile(0.25)

median = df[column].median()

Q3 = df[column].quantile(0.75)

max\_value = df[column].max()

range\_value = max\_value - min\_value

IQR = Q3 - Q1

statistics[column] = {

'Minimum Value': min\_value,

'Q1 (25th percentile)': Q1,

'Median (50th percentile)': median,

'Q3 (75th percentile)': Q3,

'Maximum Value': max\_value,

'Range': range\_value,

'Interquartile Range (IQR)': IQR

}

return statistics

**# Calculate and display the quantile statistics for all numeric columns**

quantile\_statistics = calculate\_quantile\_statistics(df)

for column, stats in quantile\_statistics.items():

print(f"\nColumn: {column}")

for stat\_name, value in stats.items():

print(f"{stat\_name}: {value}")

**# Optionally, you can save the statistics to a file**

output\_path = "Inter\_Quantile\_Output.csv"

pd.DataFrame(quantile\_statistics).T.to\_csv(output\_path)

**OUTPUT**

Column: Unnamed: 0

Minimum Value: 0

Q1 (25th percentile): 5871.25

Median (50th percentile): 11742.5

Q3 (75th percentile): 17613.75

Maximum Value: 23485

Range: 23485

Inter-quantile Range (IQR): 11742.5

Column: Clothing ID

Minimum Value: 0

Q1 (25th percentile): 861.0

Median (50th percentile): 936.0

Q3 (75th percentile): 1078.0

Maximum Value: 1205

Range: 1205

Inter-quantile Range (IQR): 217.0

Column: Age

Minimum Value: 18

Q1 (25th percentile): 34.0

Median (50th percentile): 41.0

Q3 (75th percentile): 52.0

Maximum Value: 99

Range: 81

Inter-quantile Range (IQR): 18.0

Column: Rating

Minimum Value: 1

Q1 (25th percentile): 4.0

Median (50th percentile): 5.0

Q3 (75th percentile): 5.0

Maximum Value: 5

Range: 4

Inter-quantile Range (IQR): 1.0

Column: Recommended IND

Minimum Value: 0

Q1 (25th percentile): 1.0

Median (50th percentile): 1.0

Q3 (75th percentile): 1.0

Maximum Value: 1

Range: 1

Inter-quantile Range (IQR): 0.0

Column: Positive Feedback Count

Minimum Value: 0

Q1 (25th percentile): 0.0

Median (50th percentile): 1.0

Q3 (75th percentile): 3.0

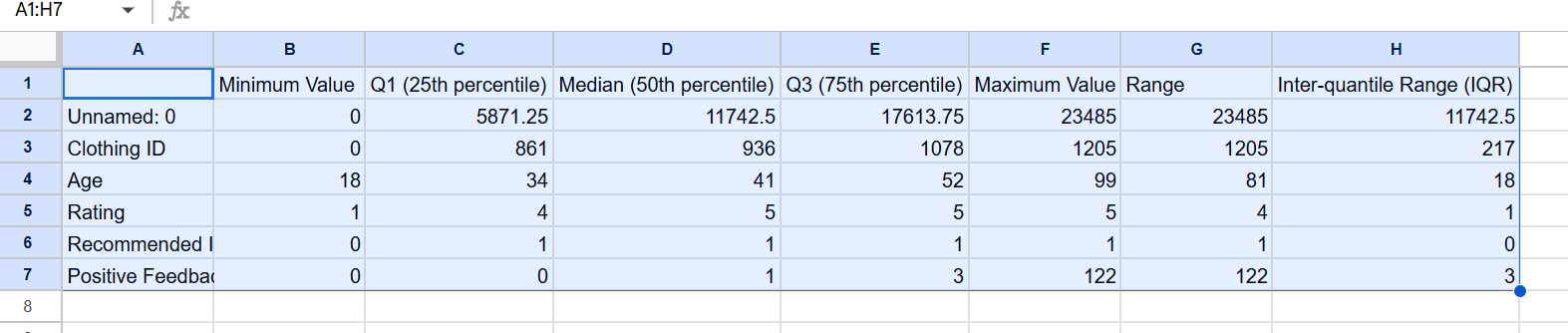
Maximum Value: 122

Range: 122

Inter-quantile Range (IQR): 3.0

Process finished with exit code 0

**Quantile Output saved as csv file**

**

# *3*.Descriptive statistics( mean, mode, standard deviation, sum, median absolute deviation, coefficient of variation, kurtosis, skewness)

**Descriptive\_Analysis.py**

**#Python Script for Descriptive Analysis**

import pandas as pd

import numpy as np

from scipy.stats import kurtosis, skew

**# Load the CSV file into a DataFrame**

file\_path = "Womens\_Clothing.csv"

df = pd.read\_csv(file\_path)

**# Function to calculate descriptive statistics**

def calculate\_descriptive\_statistics(df):

statistics = {}

for column in df.columns:

if pd.api.types.is\_numeric\_dtype(df[column]):

mean\_value = df[column].mean()

mode\_value = df[column].mode().iloc[0] if not df[column].mode().empty else np.nan

std\_dev = df[column].std()

sum\_value = df[column].sum()

mad = df[column].mad()

coeff\_var = std\_dev / mean\_value if mean\_value != 0 else np.nan

kurtosis\_value = kurtosis(df[column], nan\_policy='omit')

skewness\_value = skew(df[column], nan\_policy='omit')

statistics[column] = {

'Mean': mean\_value,

'Mode': mode\_value,

'Standard Deviation': std\_dev,

'Sum': sum\_value,

'Median Absolute Deviation': mad,

'Coefficient of Variation': coeff\_var,

'Kurtosis': kurtosis\_value,

'Skewness': skewness\_value

*}*

return statistics

**# Calculate and display the descriptive statistics for all numeric columns**

descriptive\_statistics = calculate\_descriptive\_statistics(df)

for column, stats in descriptive\_statistics.items():

print(f"\nColumn: {column}")

for stat\_name, value in stats.items():

print(f"{stat\_name}: {value}")

**# Optionally, you can save the statistics to a file**

output\_path = "Descriptive\_Analysis\_Output.csv"

pd.DataFrame(descriptive\_statistics).T.to\_csv(output\_path)

**OUTPUT**

Column: Unnamed: 0

Mean: 11742.5

Mode: 0

Standard Deviation: 6779.968547124684

Sum: 275784355

Median Absolute Deviation: 5871.5

Coefficient of Variation: 0.5773871447412974

Kurtosis: -1.2000000043510406

Skewness: 0.0

Column: Clothing ID

Mean: 918.1187090181385

Mode: 1078

Standard Deviation: 203.2989797220474

Sum: 21562936

Median Absolute Deviation: 107.0

Coefficient of Variation: 0.22142994988029482

Kurtosis: 5.180911656457766

Skewness: -2.087502669232274

Column: Age

Mean: 43.198543813335604

Mode: 39

Standard Deviation: 12.279543615591493

Sum: 1014561

Median Absolute Deviation: 8.0

Coefficient of Variation: 0.2842582765903497

Kurtosis: -0.11205236979496735

Skewness: 0.5255809358950211

Column: Rating

Mean: 4.196031678446734

Mode: 5

Standard Deviation: 1.1100307198243897

Sum: 98548

Median Absolute Deviation: 0.0

Coefficient of Variation: 0.2645429789117549

Kurtosis: 0.8037092866669644

Skewness: -1.3134452114806563

Column: Recommended IND

Mean: 0.8223622583666865

Mode: 1

Standard Deviation: 0.38221563891455684

Sum: 19314

Median Absolute Deviation: 0.0

Coefficient of Variation: 0.4647776998833635

Kurtosis: 0.8454434366260339

Skewness: -1.6868442241730668

Column: Positive Feedback Count

Mean: 2.535936302478072

Mode: 0

Standard Deviation: 5.7022015020339385

Sum: 59559

Median Absolute Deviation: 1.0

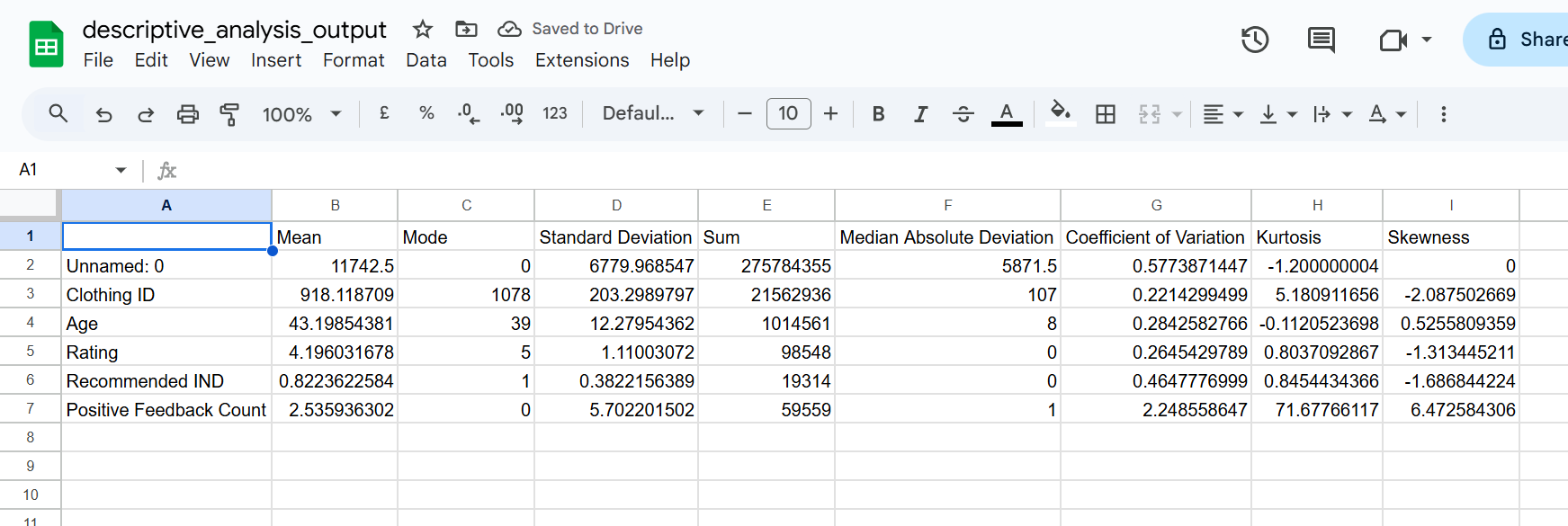
Coefficient of Variation: 2.248558647337415

Kurtosis: 71.67766116867968

Skewness: 6.472584305808119

Process finished with exit code 0

***Descriptive Analysis Output saved as a csv file***

**

# *4*.Most frequent values

**Most\_Frequent\_Values.py**

#Python Script for finding most frequent values in the given csv file

import pandas as pd

# Load the CSV file into a DataFrame

file\_path = "Womens\_Clothing.csv"

df = pd.read\_csv(file\_path)

# Function to find most frequent values

def most\_frequent\_values(df):

most\_frequent = {}

for column in df.columns:

most\_frequent\_value = df[column].mode().iloc[0] if not df[column].mode().empty else None

most\_frequent[column] = most\_frequent\_value

return most\_frequent

# Calculate and display the most frequent values for each column

most\_frequent = most\_frequent\_values(df)

for column, value in most\_frequent.items():

print(f"Column: {column}, Most Frequent Value: {value}")

# Optionally, you can save the results to a CSV file

output\_path = "Most\_Frequent\_values\_Output.csv"

pd.DataFrame(most\_frequent.items(), columns=['Column', 'Most Frequent Value']).to\_csv(output\_path, index=False)

**OUTPUT**

Column: Unnamed: 0, Most Frequent Value: 0

Column: Clothing ID, Most Frequent Value: 1078

Column: Age, Most Frequent Value: 39

Column: Title, Most Frequent Value: Love it!

Column: Review Text, Most Frequent Value: Perfect fit and i've gotten so many compliments. i buy all my suits from here now!

Column: Rating, Most Frequent Value: 5

Column: Recommended IND, Most Frequent Value: 1

Column: Positive Feedback Count, Most Frequent Value: 0

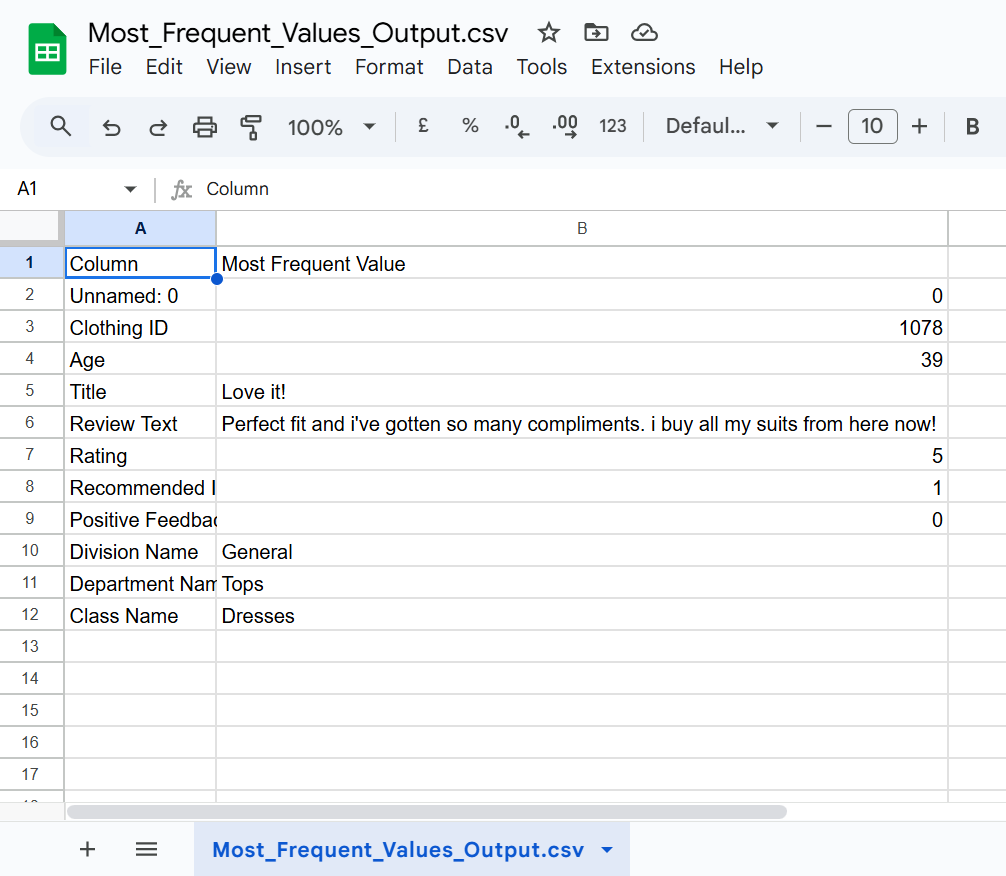
Column: Division Name, Most Frequent Value: General

Column: Department Name, Most Frequent Value: Tops

Column: Class Name, Most Frequent Value: Dresses

Process finished with exit code 0

**Most Frequent Values saved as a csv file**

******

# *5*.Histogram

**Histogram.py**

#Python Script for Histogram Graph

import pandas as pd

import matplotlib.pyplot as plt

# Load the CSV file into a DataFrame

file\_path = "Womens\_Clothing.csv"

df = pd.read\_csv(file\_path)

# Plot histograms for all numeric columns

df.hist(figsize=(10, 8), bins=30, edgecolor='black')

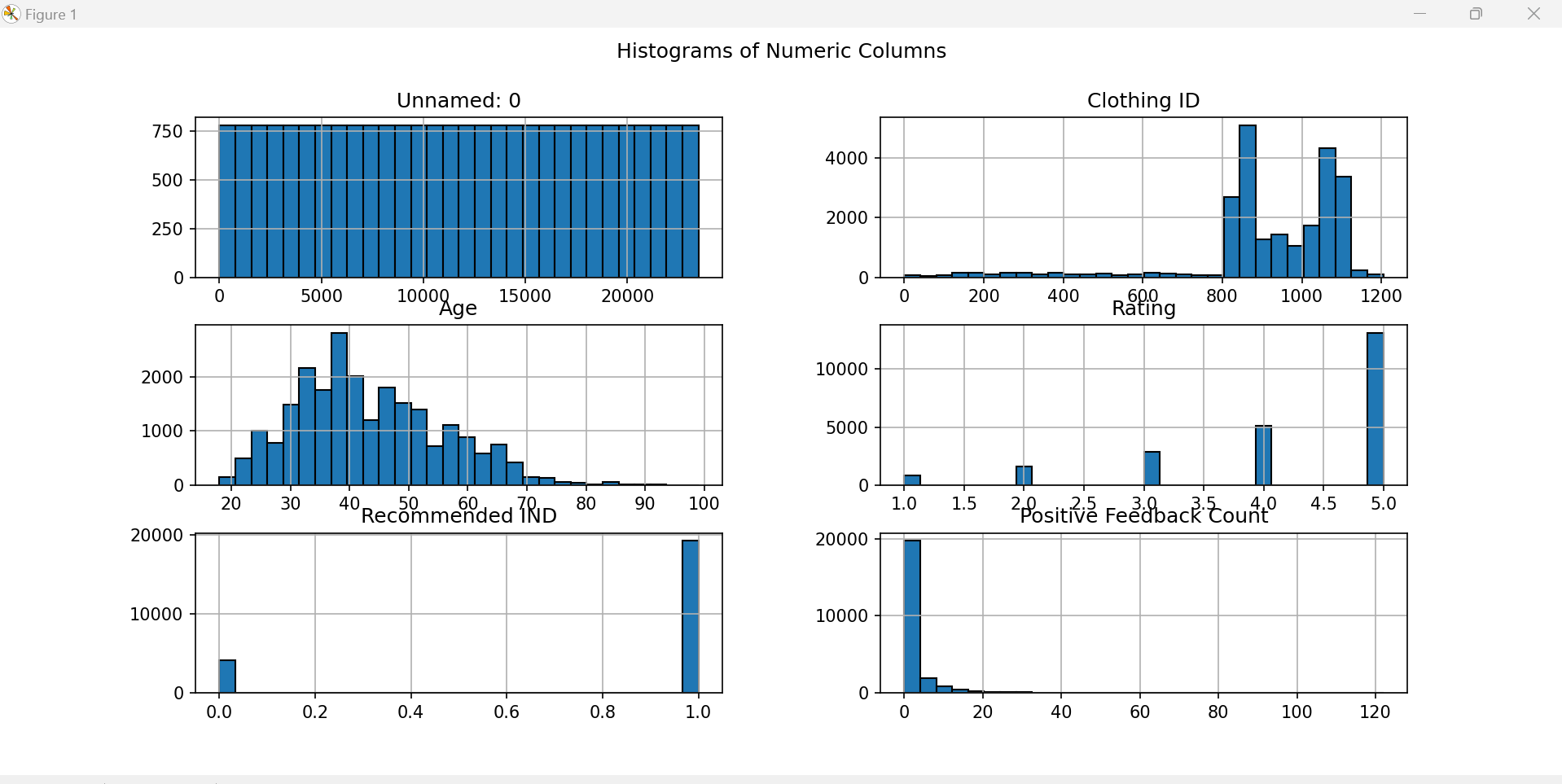
# Set overall title for the histograms

plt.suptitle('Histograms of Numeric Columns', fontsize=16)

# Display the plot

plt.show()

**Output**

******

# *6*.Correlations (highlighting of highly correlated variables, Spearman, Pearson and Kendall matrices)

**Correlations.py**

# Import the necessary library

import pandas as pd

# Define the file path to the dataset

file\_path = "Womens\_Clothing.csv"

# Load the dataset into a pandas DataFrame

df = pd.read\_csv(file\_path)

# Select only the numeric columns from the DataFrame

numeric\_df = df.select\_dtypes(include='number')

# Calculate the correlation matrices using different methods

pearson\_corr = numeric\_df.corr(method='pearson') # Pearson correlation

spearman\_corr = numeric\_df.corr(method='spearman') # Spearman correlation

kendall\_corr = numeric\_df.corr(method='kendall') # Kendall correlation

# Function to highlight highly correlated values in the correlation matrix

def highlight\_highly\_correlated(corr\_matrix, threshold=0.8):

return corr\_matrix.applymap(lambda x: 'background-color: yellow' if abs(x) >= threshold else '')

# Apply the highlighting function to the correlation matrices

highlighted\_pearson = pearson\_corr.style.apply(highlight\_highly\_correlated, threshold=0.8, axis=None)

highlighted\_spearman = spearman\_corr.style.apply(highlight\_highly\_correlated, threshold=0.8, axis=None)

highlighted\_kendall = kendall\_corr.style.apply(highlight\_highly\_correlated, threshold=0.8, axis=None)

# Save the correlation matrices to CSV files

pearson\_corr.to\_csv("pearson\_correlation.csv")

spearman\_corr.to\_csv("spearman\_correlation.csv")

kendall\_corr.to\_csv("kendall\_correlation.csv")

# Save the highlighted correlation matrices to HTML files

highlighted\_pearson.to\_html("highlighted\_pearson\_correlation.html")

highlighted\_spearman.to\_html("highlighted\_spearman\_correlation.html")

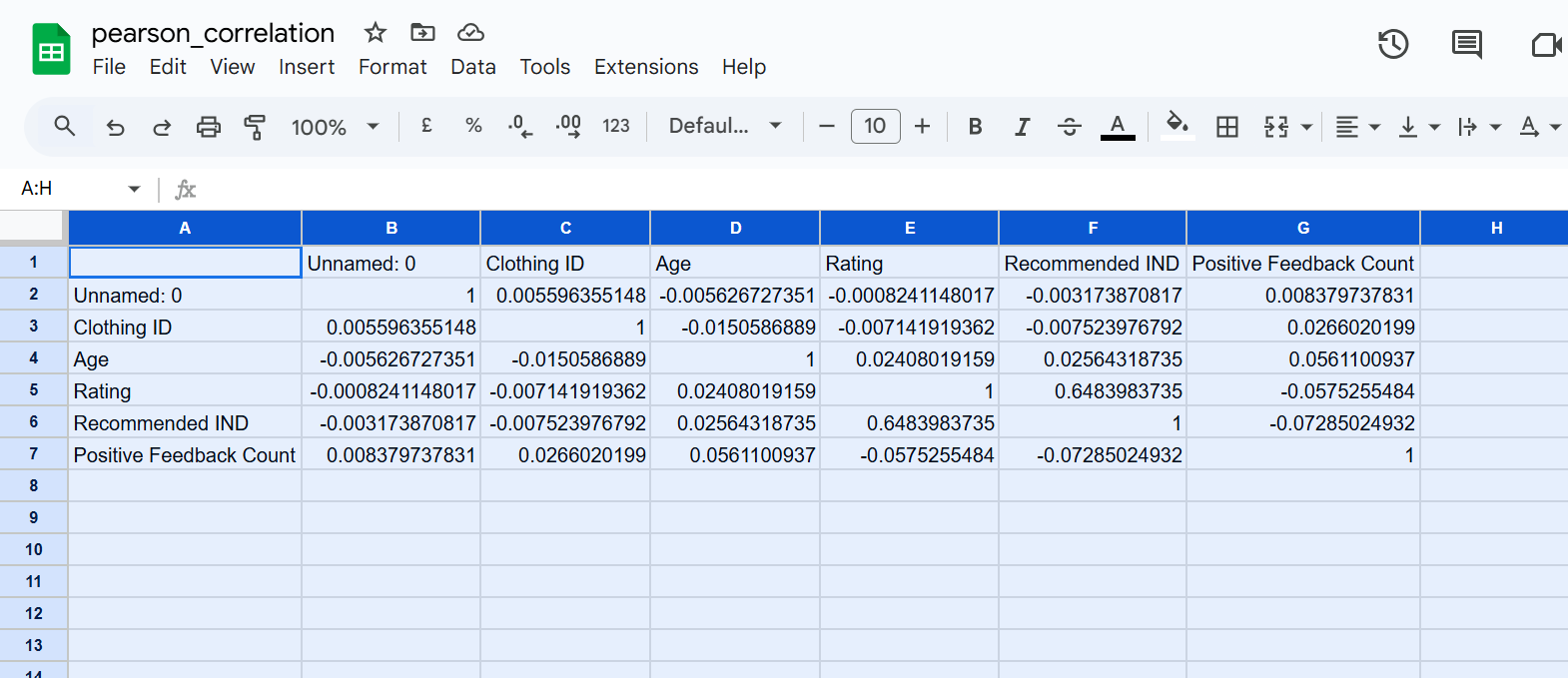
highlighted\_kendall.to\_html("highlighted\_kendall\_correlation.html")

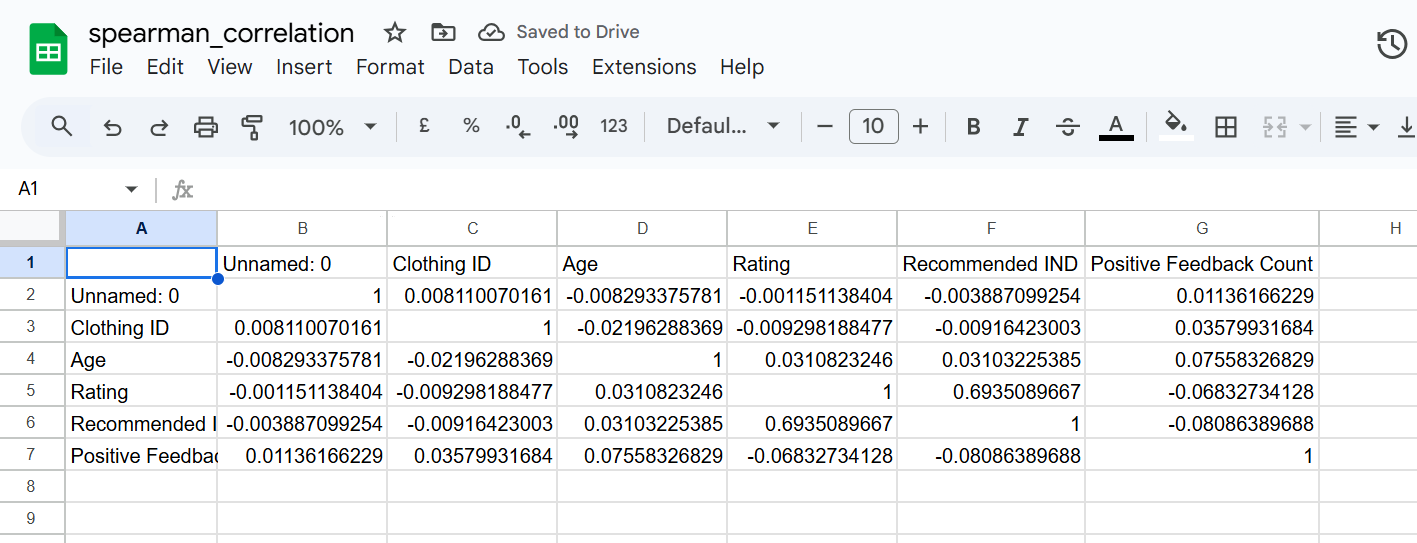
# Print a success message

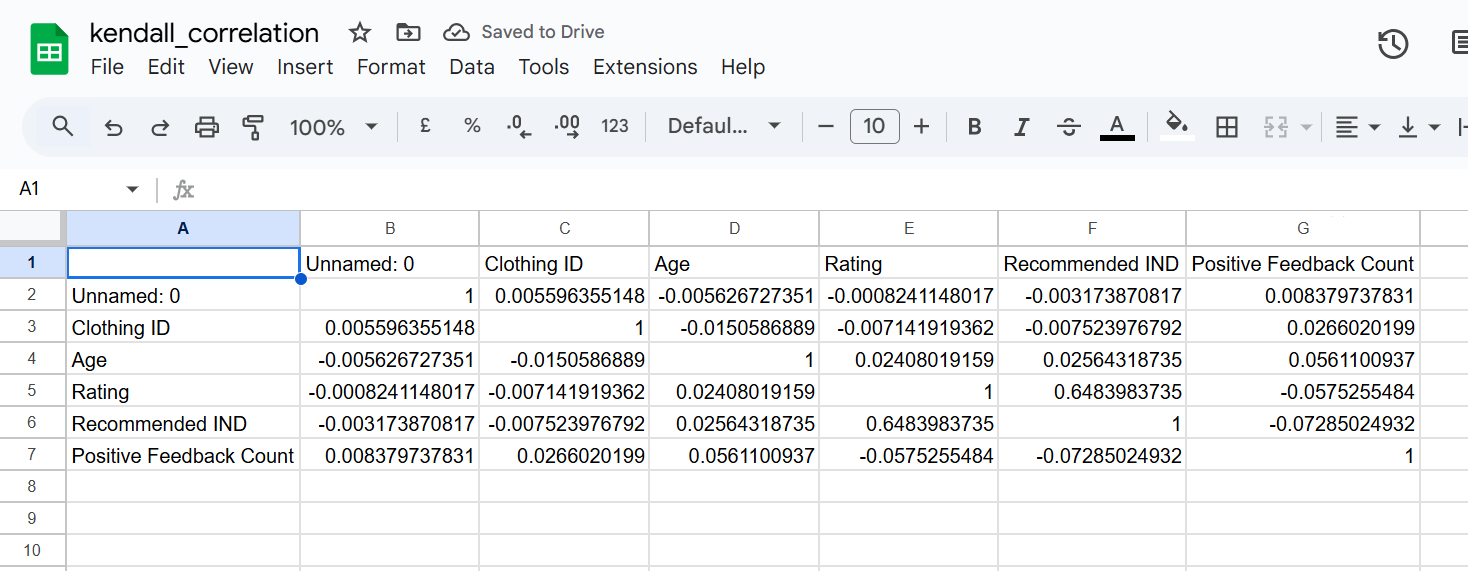
print("Correlation matrices and highlighted versions saved successfully.")

**OUTPUT**

**All the three correlations are saved as csv as well as html files.**

******

******

******

Correlation matrices and highlighted versions saved successfully.

Process finished with exit code 0

# *7*.Missing values (*matrix, count, heatmap and dendrogram of missing values)*

**Missing\_Values.py**

# Import necessary libraries

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

import missingno as msno

# Define the file path to the dataset

file\_path = "Womens\_Clothing.csv"

# Load the dataset into a pandas DataFrame

df = pd.read\_csv(file\_path)

# Calculate the count of missing values in each column

missing\_values\_count = df.isnull().sum()

# Print the count of missing values for each column

print("Missing Values Count:\n", missing\_values\_count)

# Plot a heatmap to visualize the locations of missing values

plt.figure(figsize=(12, 6))

sns.heatmap(df.isnull(), cbar=False, cmap='viridis', yticklabels=False)

plt.title('Heatmap of Missing Values')

plt.show()

# Plot a matrix to visualize the missing values using missingno

msno.matrix(df, figsize=(12, 6))

plt.title('Missing Values Matrix')

plt.show()

# Plot a dendrogram to visualize the hierarchical clustering of the missing values

msno.dendrogram(df)

plt.title('Dendrogram of Missing Values')

plt.show()

# Save the count of missing values to a CSV file

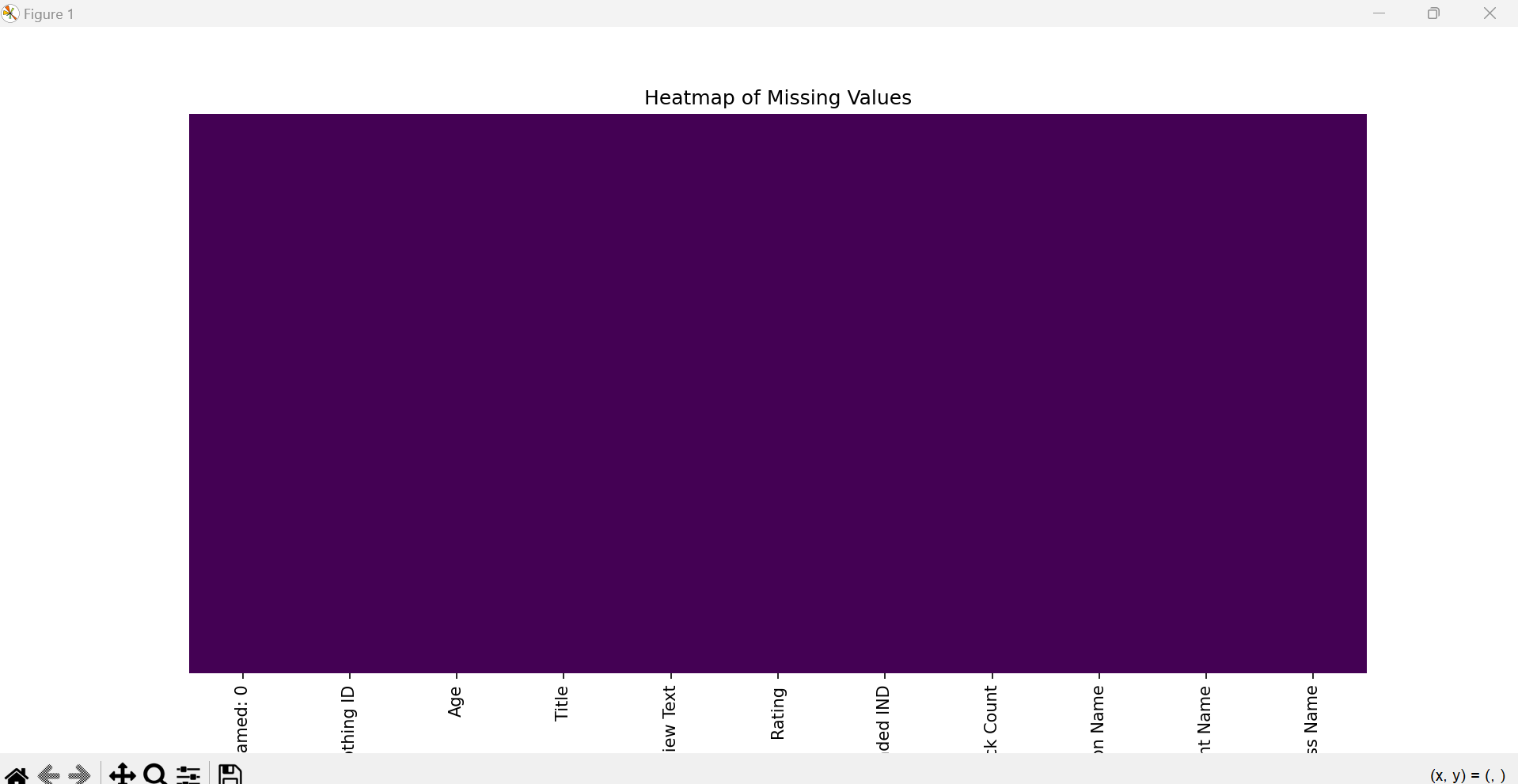
missing\_values\_count.to\_csv("missing\_value\_matrix\_output.csv", header=["Missing Values"])

# Print a success message

print("Missing values analysis completed and visualized successfully.")

**Output**

**Heat Map for Missing Values**

******

**Missing Values Count:**

Unnamed: 0 0

Clothing ID 0

Age 0

Title 0

Review Text 0

Rating 0

Recommended IND 0

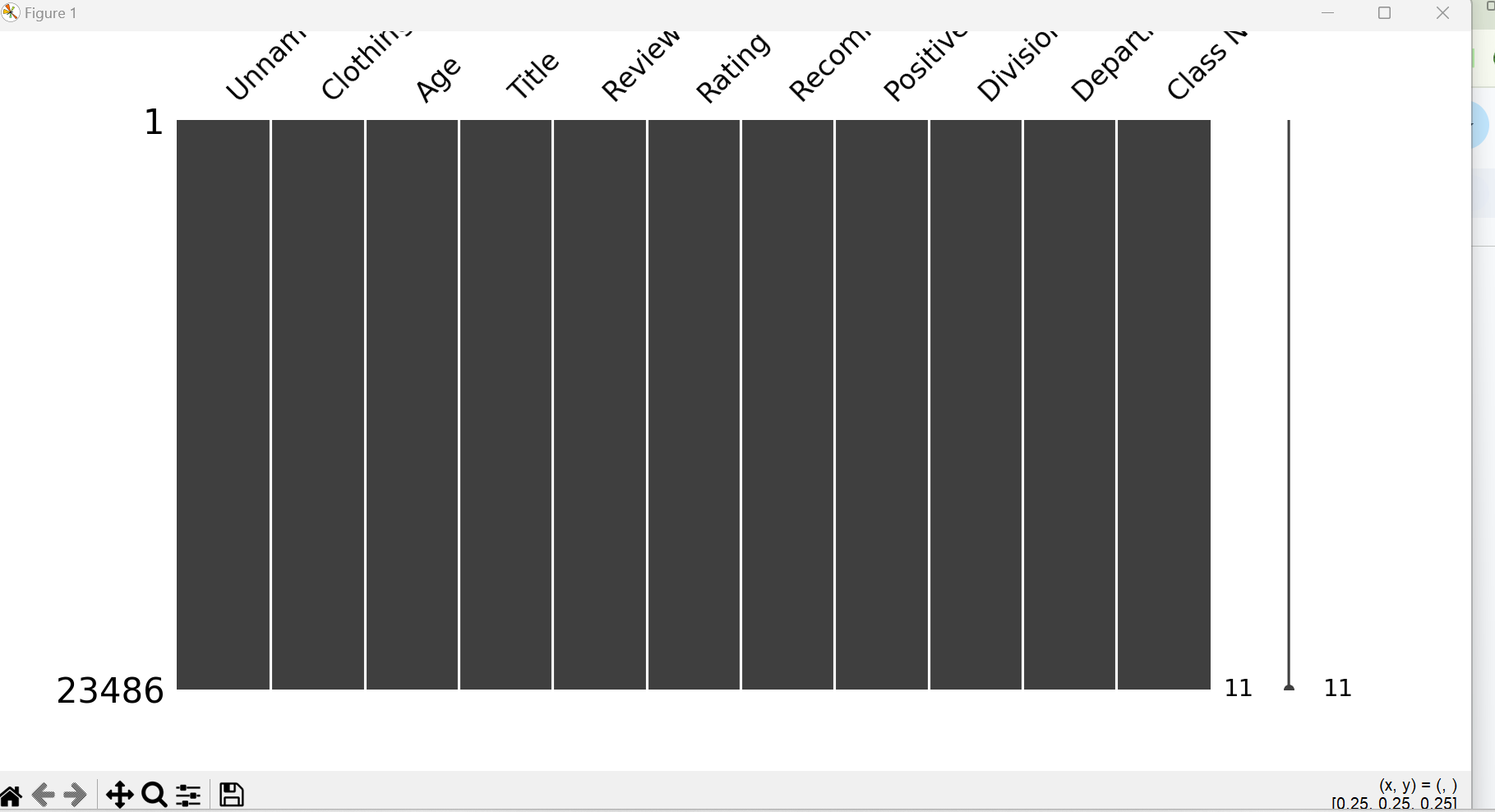
Positive Feedback Count 0

Division Name 0

Department Name 0

Class Name 0

dtype: int64

******

# 5.Data Modelling

# Calculated Measures

**Calculated Measures used in this project:**

**1.Average Age = AVERAGE(Fact\_Womens\_Clothing[Age])**

**2.Average Age by Department =**

**AVERAGEX(**

**SUMMARIZE(**

**'Fact\_Womens\_Clothing',**

**'Fact\_Womens\_Clothing'[Department Name],**

**"AvgAge", AVERAGE('Fact\_Womens\_Clothing'[Age])**

**),**

**[AvgAge]**

**)**

**3.Average Rating by Department =**

**AVERAGEX(**

**SUMMARIZE(**

**'Fact\_Womens\_Clothing',**

**'Fact\_Womens\_Clothing'[Department Name],**

**"AvgRating", AVERAGE('Fact\_Womens\_Clothing'[Rating])**

**),**

**[AvgRating]**

**)**

**4.Average Rating by Division =**

**AVERAGEX(**

**SUMMARIZE('Fact\_Womens\_Clothing', 'Fact\_Womens\_Clothing'[Division Name], "AvgRating", AVERAGE('Fact\_Womens\_Clothing'[Rating])),**

**[AvgRating]**

**)**

**5.Average Review Length = AVERAGE(Fact\_Womens\_Clothing[Review Length])**

**6.Avg Recommendation Rate =**

**DIVIDE(**

**COUNTAX(**

**FILTER('Fact\_Womens\_Clothing', 'Fact\_Womens\_Clothing'[Recommended IND] = 1),**

**'Fact\_Womens\_Clothing'[Review Text]**

**),**

**COUNT('Fact\_Womens\_Clothing'[Review Text]),**

**0**

**)**

**7.Customer Satisfaction Score = AVERAGE('Fact\_Womens\_Clothing'[Rating])**

**8.Max = 1**

**9.Negative Sentiment = CALCULATE(COUNTROWS('Fact\_Womens\_Clothing'), 'Fact\_Womens\_Clothing'[Sentiment] = "Negative")**

**)**

**12.Negative Sentiment Count by Department = CALCULATE(COUNTROWS('Fact\_Womens\_Clothing'), 'Fact\_Womens\_Clothing'[Sentiment Category] = "Negative")**

**13.Negative Sentiment Rate = DIVIDE([Negative Sentiment], [Total Reviews], 0)**

**14.Negative Sentiment Rate by Department = DIVIDE([Negative Sentiment Count by Department], [Total Reviews by Department], 0)**

**15.Neutral Sentiment = CALCULATE(COUNTROWS('Fact\_Womens\_Clothing'), 'Fact\_Womens\_Clothing'[Sentiment] = "Neutral")**

**16.Positive Feedback Count by Department =**

**COUNTROWS(**

**FILTER(**

**'Fact\_Womens\_Clothing',**

**'Fact\_Womens\_Clothing'[Positive Feedback Count] > 0**

**)**

**)**

**17.Positive Feedback Rate =**

**DIVIDE (**

**COUNTROWS(FILTER('Fact\_Womens\_Clothing', 'Fact\_Womens\_Clothing'[Positive Feedback Count] > 0)),**

**COUNTROWS('Fact\_Womens\_Clothing'),**

**0**

**)**

**18.Positive Feedback Rate by Department =**

**DIVIDE(**

**[Positive Feedback Count by Department],**

**[Total Reviews by Department],**

**0**

**)**

**19.Positive Sentiment = CALCULATE(COUNTROWS('Fact\_Womens\_Clothing'), 'Fact\_Womens\_Clothing'[Sentiment] = "Positive")**

**22.Positive Sentiment Count by Department = CALCULATE(COUNTROWS('Fact\_Womens\_Clothing'), 'Fact\_Womens\_Clothing'[Sentiment Category] = "Positive")**

**23.Positive Sentiment Rate = DIVIDE([Positive Sentiment], [Total Reviews], 0)**

**24.Positive Sentiment Rate by Department = DIVIDE([Positive Sentiment Count by Department], [Total Reviews by Department], 0)**

**25.Recommendation Rate =**

**DIVIDE (**

**COUNTROWS(FILTER('Fact\_Womens\_Clothing', 'Fact\_Womens\_Clothing'[Recommended IND] = 1)),**

**COUNTROWS('Fact\_Womens\_Clothing'),**

**0**

**)**

**26.Reviews by Division = COUNTROWS('Fact\_Womens\_Clothing')**

**27.Top Reviewed Division =**

**VAR DivisionReviewCounts =**

**SUMMARIZE(**

**'Fact\_Womens\_Clothing',**

**'Fact\_Womens\_Clothing'[Division Name],**

**"Review Count", COUNTROWS('Fact\_Womens\_Clothing')**

**)**

**VAR TopDivision =**

**TOPN(**

**1,**

**DivisionReviewCounts,**

**[Review Count],**

**DESC**

**)**

**VAR TopDivisionName =**

**MAXX(TopDivision, 'Fact\_Womens\_Clothing'[Division Name])**

**RETURN**

**TopDivisionName**

**28.Total Reviews = COUNTROWS('Fact\_Womens\_Clothing')**

**29.Total Reviews by Department =**

**COUNTROWS('Fact\_Womens\_Clothing')**

**Calculated Columns**

**1.Age Group =**

**SWITCH (**

**TRUE(),**

**'Fact\_Womens\_Clothing'[Age] < 20, "Under 20",**

**'Fact\_Womens\_Clothing'[Age] < 30, "20-29",**

**'Fact\_Womens\_Clothing'[Age] < 40, "30-39",**

**'Fact\_Womens\_Clothing'[Age] < 50, "40-49",**

**'Fact\_Womens\_Clothing'[Age] < 60, "50-59",**

**'Fact\_Womens\_Clothing'[Age] >= 60, "60+",**

**"Unknown"**

**)**

**2.Negative\_Reviews = IF(CONTAINSSTRING([Cleaned Review Text], "negative"), 1, 0)**

**3.Positive\_Reviews = IF(CONTAINSSTRING([Cleaned Review Text], "positive"), 1, 0)**

**4.Review Length = LEN('Fact\_Womens\_Clothing'[Cleaned Review Text])**

**5.Sentiment Category =**

**SWITCH (**

**TRUE(),**

**[Sentiment Score] > 0, "Positive",**

**[Sentiment Score] < 0, "Negative",**

**[Sentiment Score] = 0, "Neutral",**

**BLANK()**

**)**

**6.Word Count = LEN([Cleaned Review Text]) - LEN(SUBSTITUTE([Cleaned Review Text], " ", "")) + 1**

**Correlation Variable**

**1.11.Negative Sentiment and Sentiment Score correlation for Department Name =**

**VAR \_\_CORRELATION\_TABLE = VALUES('Fact\_Womens\_Clothing'[Department Name])**

**VAR \_\_COUNT =**

**COUNTX(**

**KEEPFILTERS(\_\_CORRELATION\_TABLE),**

**CALCULATE(**

**[Negative Sentiment]**

**\* SUM('Fact\_Womens\_Clothing'[Sentiment Score])**

**)**

**)**

**VAR \_\_SUM\_X =**

**SUMX(**

**KEEPFILTERS(\_\_CORRELATION\_TABLE),**

**CALCULATE([Negative Sentiment])**

**)**

**VAR \_\_SUM\_Y =**

**SUMX(**

**KEEPFILTERS(\_\_CORRELATION\_TABLE),**

**CALCULATE(SUM('Fact\_Womens\_Clothing'[Sentiment Score]))**

**)**

**VAR \_\_SUM\_XY =**

**SUMX(**

**KEEPFILTERS(\_\_CORRELATION\_TABLE),**

**CALCULATE(**

**[Negative Sentiment]**

**\* SUM('Fact\_Womens\_Clothing'[Sentiment Score]) \* 1.**

**)**

**)**

**VAR \_\_SUM\_X2 =**

**SUMX(**

**KEEPFILTERS(\_\_CORRELATION\_TABLE),**

**CALCULATE([Negative Sentiment] ^ 2)**

**)**

**VAR \_\_SUM\_Y2 =**

**SUMX(**

**KEEPFILTERS(\_\_CORRELATION\_TABLE),**

**CALCULATE(SUM('Fact\_Womens\_Clothing'[Sentiment Score]) ^ 2)**

**)**

**RETURN**

**DIVIDE(**

**\_\_COUNT \* \_\_SUM\_XY - \_\_SUM\_X \* \_\_SUM\_Y \* 1.,**

**SQRT(**

**(\_\_COUNT \* \_\_SUM\_X2 - \_\_SUM\_X ^ 2)**

**\* (\_\_COUNT \* \_\_SUM\_Y2 - \_\_SUM\_Y ^ 2)**

**)**

**)**

**2.21.Positive Sentiment and Sentiment Score correlation for Department Name =**

**VAR \_\_CORRELATION\_TABLE = VALUES('Fact\_Womens\_Clothing'[Department Name])**

**VAR \_\_COUNT =**

**COUNTX(**

**KEEPFILTERS(\_\_CORRELATION\_TABLE),**

**CALCULATE(**

**[Positive Sentiment]**

**\* SUM('Fact\_Womens\_Clothing'[Sentiment Score])**

**)**

**)**

**VAR \_\_SUM\_X =**

**SUMX(**

**KEEPFILTERS(\_\_CORRELATION\_TABLE),**

**CALCULATE([Positive Sentiment])**

**)**

**VAR \_\_SUM\_Y =**

**SUMX(**

**KEEPFILTERS(\_\_CORRELATION\_TABLE),**

**CALCULATE(SUM('Fact\_Womens\_Clothing'[Sentiment Score]))**

**)**

**VAR \_\_SUM\_XY =**

**SUMX(**

**KEEPFILTERS(\_\_CORRELATION\_TABLE),**

**CALCULATE(**

**[Positive Sentiment]**

**\* SUM('Fact\_Womens\_Clothing'[Sentiment Score]) \* 1.**

**)**

**)**

**VAR \_\_SUM\_X2 =**

**SUMX(**

**KEEPFILTERS(\_\_CORRELATION\_TABLE),**

**CALCULATE([Positive Sentiment] ^ 2)**

**)**

**VAR \_\_SUM\_Y2 =**

**SUMX(**

**KEEPFILTERS(\_\_CORRELATION\_TABLE),**

**CALCULATE(SUM('Fact\_Womens\_Clothing'[Sentiment Score]) ^ 2)**

**)**

**RETURN**

**DIVIDE(**

**\_\_COUNT \* \_\_SUM\_XY - \_\_SUM\_X \* \_\_SUM\_Y \* 1.,**

**SQRT(**

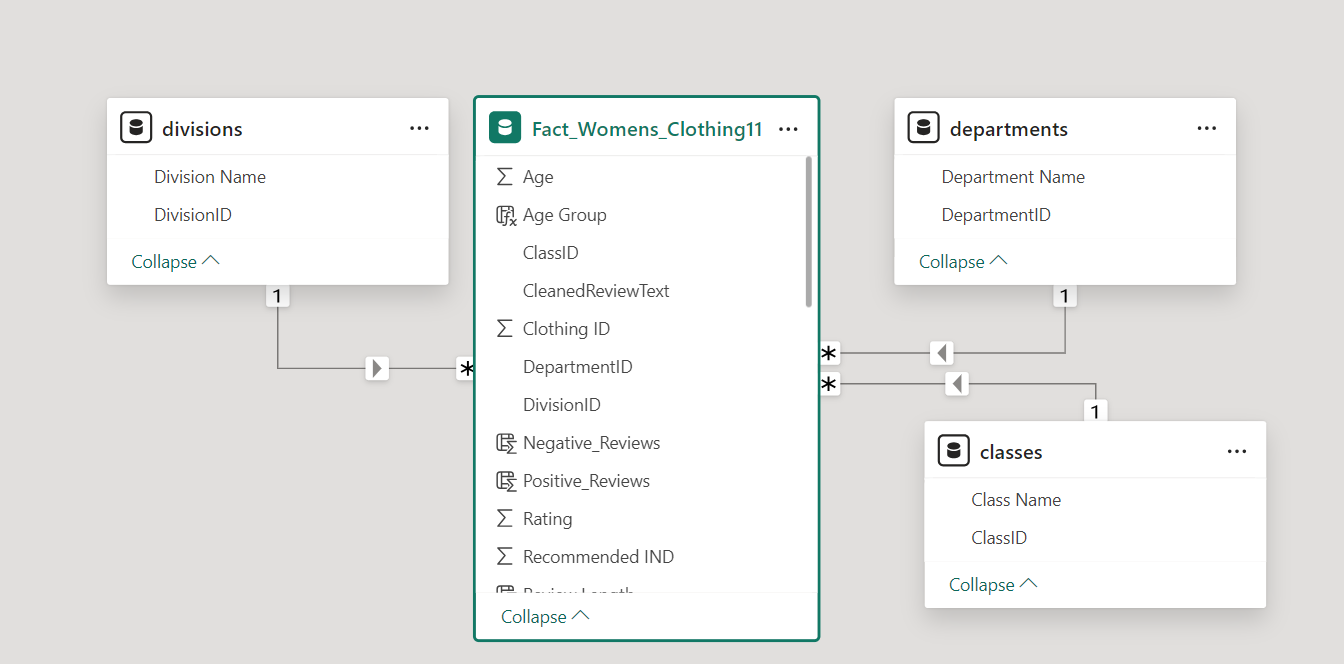
**(\_\_COUNT \* \_\_SUM\_X2 - \_\_SUM\_X ^ 2)**

**\* (\_\_COUNT \* \_\_SUM\_Y2 - \_\_SUM\_Y ^ 2)**

**)**

**)**

**Data Modelling(Snowflake Model)**

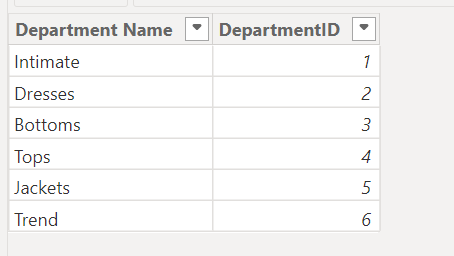
****

**Dimension Tables**

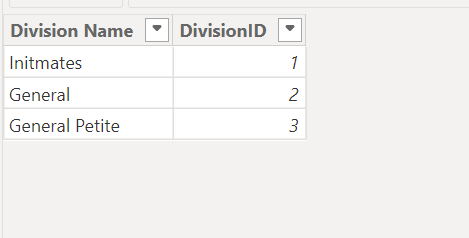
**1.Dim\_Classes**

****

**2.Dim\_Department**

****

**3.Dim\_Division**

****

**Fact Table**

**Fact\_WomensClothing**

**6.Exploratory Data Analysis (EDA)**

**Page 1: Sales Performance Home Page**

****

**Slicers:**

### **Key Insights from Visualizations**

#### **Slicers for Age, Division Name, Department Name, and Class Name**

**Purpose:**

* To allow dynamic filtering of data based on age, division name, department name, and class name.
* These slicers help in drilling down into specific subsets of the data for more detailed analysis.

**Insight:**

* Slicers make it easier to understand how different demographic segments interact with various departments and classes, offering a customizable view of the data.

#### **Table for Top 3 Departments with High Positive Feedback Rating**

**Purpose:**

* To identify departments that receive the highest positive feedback from customers.

**Insight:**

1. General Tops: Consistently receives high positive feedback, indicating strong customer satisfaction with products in this department.
2. General Dresses: Also ranks high in positive feedback, suggesting that customers are very pleased with the quality and variety of dresses offered.
3. General Petit Dresses: Shows significant positive feedback, highlighting the department's success in meeting customer expectations.

#### **Cards for Customer Satisfaction Score, Average Age, Positive Feedback Rate, Total Reviews, Neutral Sentiment, and Top Reviewed Division**

**Purpose:**

* To provide key metrics at a glance.

**Insights:**

1. **Customer Satisfaction Score:** Reflects overall satisfaction levels. A high score indicates that most customers are happy with their purchases.
2. **Average Age:** Shows the average age of reviewers, helping to identify the primary demographic engaging with the products.
3. **Positive Feedback Rate:** Demonstrates the proportion of reviews that are positive, indicating overall product approval.
4. **Total Reviews:** Provides a count of all reviews, reflecting customer engagement.
5. **Neutral Sentiment:** Measures the proportion of neutral reviews, giving a sense of ambiguity in customer feedback.
6. **Top Reviewed Division:** Identifies which division receives the most reviews, highlighting areas of high customer interest and engagement.

#### **Pie Chart for Sentiment Distribution**

**Purpose:**

* To visualize the distribution of sentiments (positive, neutral, negative) across all reviews.

**Insight:**

* A majority of the reviews are positive, with a smaller proportion being neutral or negative. This indicates that overall, customers are satisfied with their purchases, but there are areas for improvement to reduce negative feedback.

#### **Column Chart for Reviews by Division Names**

**Purpose:**

* To show the number of reviews received by each division.

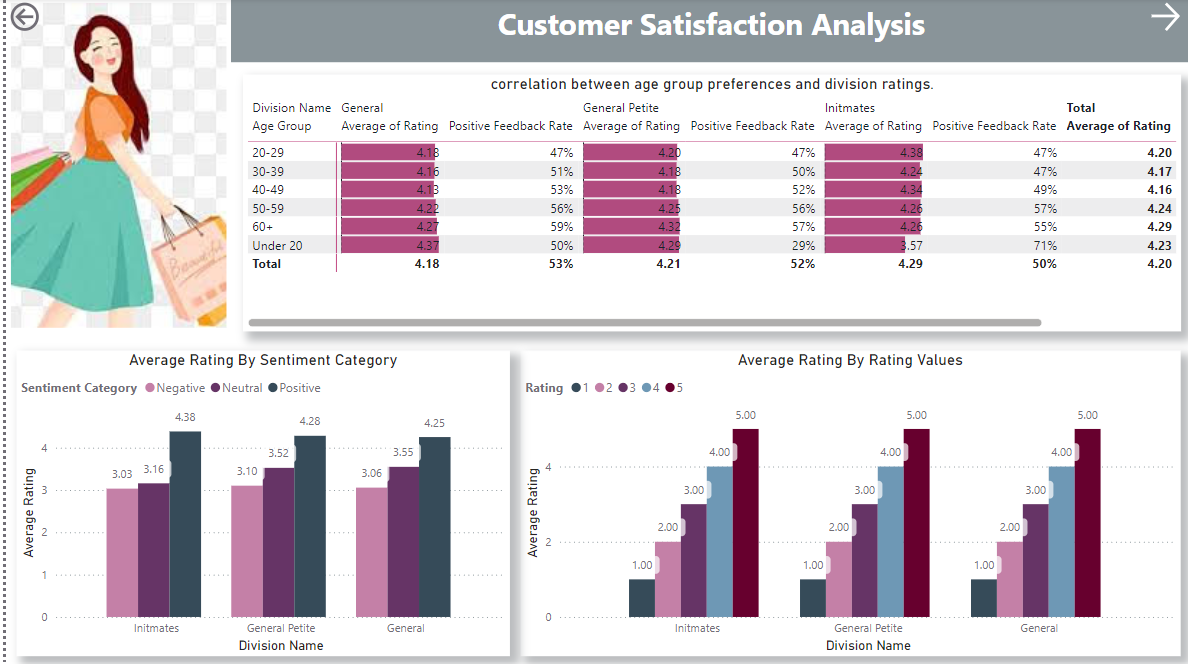
**Insight:**

* Certain divisions, such as Tops and Dresses, receive a significantly higher number of reviews compared to others. This highlights popular product categories that attract more customer attention and engagement.

### **Overall Key Insights**

* **High Customer Satisfaction:** The customer satisfaction score and positive feedback rate suggest that most customers are happy with their purchases. Focusing on maintaining product quality in top-performing departments like Tops, Dresses, and Outerwear will sustain this satisfaction.
* **Targeted Marketing:** The average age of reviewers helps identify the primary demographic, enabling more targeted marketing efforts. Engaging with this demographic through personalized campaigns can enhance customer loyalty.
* **Improvement Areas:** The sentiment distribution and neutral sentiment card reveal that there are areas needing attention. Addressing issues highlighted in neutral and negative reviews can further enhance customer satisfaction.
* **Engagement Insights:** The total reviews and top reviewed division metrics show high customer engagement in certain divisions. Leveraging this interest through promotions and new product launches can drive sales in these areas.
* **Popularity of Certain Divisions:** The column chart for reviews by division names indicates which divisions are most popular among customers. This information can guide inventory and product development decisions to focus on high-demand categories.

### **Page 2: Customer Satisfaction Analysis(Rating Analysis)**



### **Correlation Matrix: Age Group Performance and Division Rating**

**Purpose:** To analyze the relationship between different age groups and the ratings given to various divisions. This helps in understanding how different age groups perceive products in different divisions.

**Visualization:** Correlation Matrix

**Description:**

* Matrix Table: Displays the correlation coefficients between age groups and division ratings.
* Insights:
  + Higher correlations between certain age groups and specific divisions indicate a stronger relationship between the age group and their ratings of the division.
  + Identifies key age groups that influence ratings in particular divisions.

**Key Insights:**

* 30-39 and above Age Group: Shows a strong positive correlation with high ratings in the Dresses and Outerwear divisions.
* Under-20 Age Group: Displays a weaker correlation with division ratings, indicating less influence on overall ratings.

### **Clustered Bar Chart: Average Rating by Sentiment Category**

**Purpose:** To visualize the average rating provided by customers, categorized by their sentiment (positive, neutral, negative). This helps to understand how sentiments affect ratings.

**Visualization:** Clustered Bar Chart

**Description:**

* Axis: Sentiment Category (Positive, Neutral, Negative)
* Values: Average Rating
* Insights:
  + Positive sentiment reviews generally have higher average ratings.
  + Neutral sentiment reviews provide moderate ratings.
  + Negative sentiment reviews typically result in lower ratings.

**Key Insights:**

* Positive Sentiment: Associated with the highest average ratings, reflecting customer satisfaction.
* Neutral Sentiment: Shows moderate ratings, indicating room for improvement in certain areas.
* Negative Sentiment: Corresponds to the lowest average ratings, highlighting areas that need attention and improvement.

### **Bar Chart: Average Rating by Rating Values**

Purpose: To display the distribution of average ratings given by customers, categorized by rating values (1-5 stars). This helps in identifying the most common rating values and overall customer satisfaction.

Visualization: Bar Chart

Description:

* Axis: Rating Values (1, 2, 3, 4, 5)
* Values: Average Rating
* Insights:
  + The frequency of each rating value shows the overall distribution of ratings.
  + Helps in understanding the common rating trends among customers.

**Key Insights:**

* 5 Stars: The highest frequency, indicating that a large proportion of customers are extremely satisfied with their purchases.
* 4 Stars: Also shows a high frequency, reflecting general satisfaction but with minor areas for improvement.
* 1-3 Stars: Lower frequencies, suggesting less common but notable dissatisfaction among some customers.

### **Summary of Insights and Recommendations**

**Overall Insights:**

1. **Correlation Analysis:**
   * The 30-39 age group significantly influences ratings in divisions like Dresses and Outerwear.
   * The under-20 age group has less impact on division ratings.
2. **Sentiment and Ratings:**
   * Positive sentiments are strongly associated with higher ratings.
   * Negative sentiments highlight areas that need improvement.
3. **Rating Distribution:**
   * A majority of customers provide high ratings (4-5 stars), indicating general satisfaction.
   * Lower ratings (1-3 stars) need to be addressed to enhance overall customer experience.

**Recommendations:**

1. **Targeted Improvements:**
   * Focus on areas receiving negative feedback to improve product quality and customer satisfaction.
2. **Marketing Strategies:**
   * Leverage positive feedback from the 30-39 age group in targeted marketing campaigns.
3. **Product Development:**
   * Address common issues highlighted in neutral and negative reviews to reduce dissatisfaction.

### **Page 3: Sentiment Analysis**



Page 3: Sentiment Analysis

### **Key Gauge: Correlation Coefficient**

Purpose: To display the correlation coefficient between positive sentiment, sentiment score, and department category, highlighting the strength of the relationship.

**Visualization: Key Gauge**

**Description:**

* Key Gauge: Shows the correlation coefficient value.
* Insights:
  + A higher correlation coefficient indicates a strong relationship between positive sentiment, sentiment score, and department category.
  + Helps in identifying which departments are most positively perceived by customers.

**Key Insights:**

* Positive Sentiment: Departments with high positive sentiment scores exhibit a strong correlation, indicating customer satisfaction.
* Negative Sentiment: Departments with high negative sentiment scores show a different correlation pattern, highlighting areas for improvement.

### **Scatter Plot: Sentiment Score vs. Positive Sentiment by Department Category**

**Purpose:** To visualize the relationship between sentiment score and positive sentiment across different department categories.

**Visualization: Scatter Plot**

**Description:**

* **X-Axis:** Sentiment Score
* **Y-Axis:** Positive Sentiment
* **Color/Shape:** Department Category
* **Insights:**
  + Clusters of points can indicate departments with consistent positive sentiment and high sentiment scores.
  + Outliers may reveal departments that differ significantly in customer perception.

**Key Insights:**

* **Clusters:** Departments like Tops and Dresses show tight clusters with high positive sentiment and sentiment scores.
* **Outliers:** No outliers.

### **Scatter Plot: Sentiment Score vs. Negative Sentiment by Department Category**

**Purpose:** To analyze the relationship between sentiment score and negative sentiment across different department categories.

**Visualization: Scatter Plot**

**Description:**

* **X-Axis:** Sentiment Score
* **Y-Axis:** Negative Sentiment
* **Color/Shape:** Department Category
* **Insights:**
  + Identifies departments with high negative sentiment and low sentiment scores.
  + Highlights areas needing improvement to reduce negative customer feedback.

**Key Insights:**

* **High Negative Sentiment:** Departments with higher negative sentiment, indicating potential issues.
* **Focus Areas:** These departments may require targeted improvements to enhance customer satisfaction.

### **Key Gauge: Correlation Coefficient by Negative Sentiment, Sentiment Score, and Department Category**

**Purpose:** To display the correlation coefficient between negative sentiment, sentiment score, and department category.

**Visualization:** Key Gauge

**Description:**

* **Key Gauge:** Shows the correlation coefficient value for negative sentiment.
* **Insights:**
  + A higher correlation coefficient indicates a strong relationship between negative sentiment, sentiment score, and department category.
  + Helps in identifying which departments are most negatively perceived by customers.

**Key Insights:**

* **Negative Sentiment:** Departments with high negative sentiment scores exhibit a strong correlation, pinpointing areas for improvement.
* **Sentiment Score:** Departments with lower sentiment scores and higher negative sentiments need targeted intervention.

### **Summary of Insights and Recommendations**

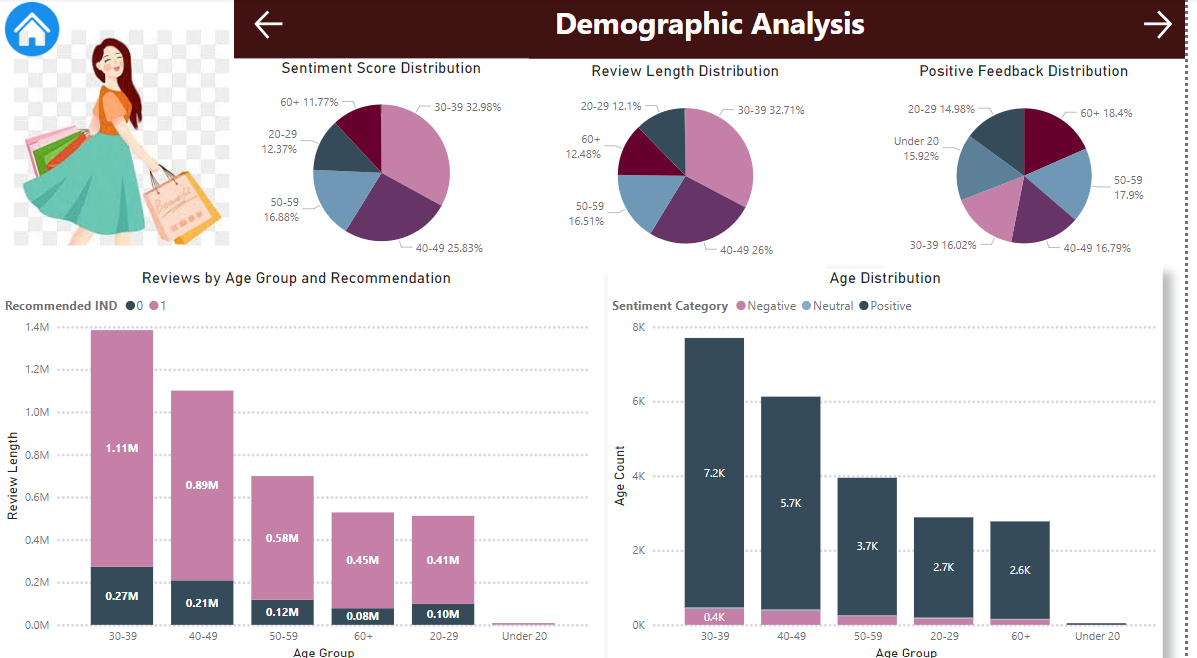
**Overall Insights:**

1. **Positive Sentiment Analysis:**
   * Departments with high positive sentiment scores show a strong correlation, indicating high customer satisfaction.
   * Key departments like Tops and Dresses consistently receive positive feedback, highlighting their success.
2. **Negative Sentiment Analysis:**
   * Departments with high negative sentiment scores show significant correlation, identifying areas needing improvement.
   * Departments with higher negative sentiment, suggesting the need for product and service enhancements.

**Recommendations:**

1. **Enhance Positive Performers:**
   * Continue to promote and invest in departments with high positive sentiment scores, such as Tops and Dresses.
2. **Address Negative Feedback:**
   * Focus on improving departments with high negative sentiment scores. Target common issues identified in feedback.
3. **Customer Engagement:**
   * Use sentiment analysis to tailor customer engagement strategies. Leverage positive sentiments in marketing and address negative feedback proactively.

### **Page 4: Demographic Analysis**



### **Pie Chart: Sentiment Score Distribution**

**Purpose**: To visualize the distribution of sentiment scores across all reviews, providing insights into the overall emotional tone of customer feedback.

**Visualization**: Pie Chart

**Description**:

* **Sections**: Represents different ranges or categories of sentiment scores (e.g., positive, neutral, negative).
* **Insights**:
  + A higher proportion of positive sentiment scores indicates overall customer satisfaction.
  + The distribution helps identify the balance between positive, neutral, and negative sentiments.

**Key Insights**:

* **Positive Sentiment Dominance**: A large portion of the pie chart is occupied by positive sentiment scores, showing that most customers are satisfied with their purchases.
* **Areas for Improvement**: Smaller sections representing neutral and negative sentiments highlight areas that need attention to improve customer satisfaction.

### **2. Pie Chart: Review Length Distribution**

**Purpose**: To show the distribution of review length by Age group helping to understand how detailed the feedback is from different age ranges.

**Visualization**: Pie Chart

**Description**:

* **Sections**: Represents different ranges of review lengths (e.g., short, medium, long).
* **Insights**:
  + Identifies which age group gives brief or detailed reviews.
  + Helps in understanding engagement levels and feedback quality.

**Key Insights**:

* **Detailed Feedback**: A significant portion of the chart is occupied by longer reviews, indicating high engagement and detailed feedback from the 30-39 Age group.
* **Brief Feedback**: Smaller sections represent shorter reviews, highlighting the need to encourage more detailed feedback from 20-29 Age Group.

### **Pie Chart: Positive Feedback Distribution**

**Purpose**: To visualize the distribution of positive feedback across different Age Ranges, showing which Age Range is giving more Positive feedback.

**Visualization**: Pie Chart

**Description**:

* **Sections**: Represents different departments.
* **Insights**:
  + Which Age Group is providing more Positive feedback.
  + Provides a clear picture of customer satisfaction by Age Group.

**Key Insights**:

* **Top Positive Feedback:** Age Range 60+ is providing more high levels of positive feedback.
* **Least Positive Feedback** :Age Range 20-29 provides the least.May need targeted improvements to enhance customer satisfaction.

### **Clustered Column Chart: Reviews by Age Group Category by Recommended IND**

**Purpose**: To analyze the distribution of reviews across different age groups, categorized by whether the product was recommended or not.

**Visualization**: Clustered Column Chart

**Description**:

* **X-Axis**: Age Group Category
* **Y-Axis**: Number of Reviews
* **Legend**: Recommended Indicator (Yes or No)
* **Insights**:
  + Shows which age groups are more likely to recommend products.
  + Highlights the engagement level of different age groups.

**Key Insights**:

* **High Engagement**: The 30-39 age group shows a high number of reviews and recommendations, indicating strong engagement and satisfaction.
* **Lower Engagement**: The under-20 age group has fewer reviews and recommendations, suggesting the need for strategies to increase their engagement.

### **Clustered Column Chart: Age Distribution Categorized by Sentiment Category**

**Purpose**: To visualize the distribution of age groups across different sentiment categories (positive, neutral, negative).

**Visualization**: Clustered Column Chart

**Description**:

* **X-Axis**: Age Group
* **Y-Axis**: Number of Reviews
* **Legend**: Sentiment Category (Positive, Neutral, Negative)
* **Insights**:
  + Identifies the sentiment trends within different age groups.
  + Helps in understanding the emotional tone of feedback from different demographics.

**Key Insights**:

* **Positive Feedback**: The 30-39 age group has the highest number of positive reviews, indicating strong satisfaction.
* **Neutral and Negative Feedback**: The under-20 age group shows a higher proportion of neutral and negative feedback, suggesting areas for improvement in catering to younger customers.

### **Summary of Insights and Recommendations**

**Overall Insights**:

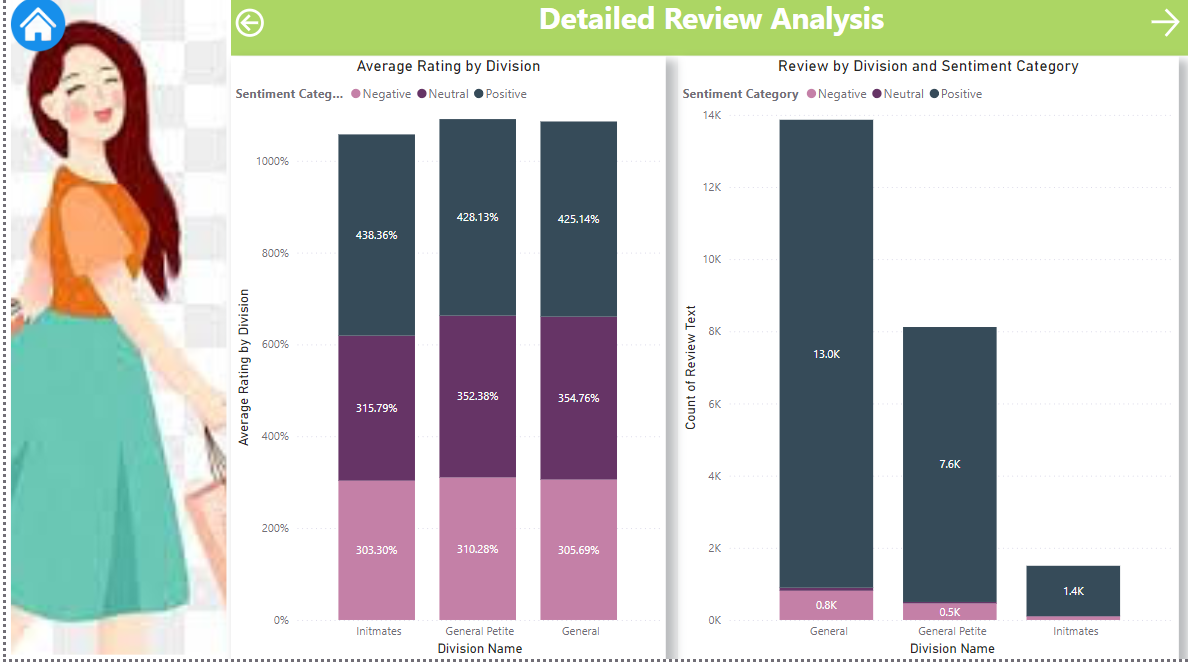
1. **Sentiment Distribution**:
   * Positive sentiments dominate, indicating overall customer satisfaction.
   * Neutral and negative sentiments highlight areas needing improvement.
2. **Review Length**:
   * Customers tend to leave detailed reviews, indicating high engagement.
   * Encouraging more detailed feedback can provide richer insights.
3. **Department Performance**:
   * Tops and Dresses departments receive the most positive feedback.
   * Some departments need improvement to boost customer satisfaction.
4. **Age Group Analysis**:
   * The 30-39 age group is highly engaged and satisfied.
   * The under-20 age group needs targeted strategies to increase engagement and satisfaction.

**Recommendations**:

1. **Enhance Positive Performers**:
   * Continue to promote and invest in departments with high positive feedback, such as Tops and Dresses.
2. **Address Negative Feedback**:
   * Focus on improving departments with higher negative sentiment scores, like Shoes.
3. **Engage Younger Customers**:
   * Develop targeted strategies to increase engagement and satisfaction among the under-20 age group.

This documentation provides a comprehensive view of the demographic analysis, highlighting key insights and actionable recommendations based on customer feedback and sentiment analysis.

### **Page 5: Detailed Review Analysis**



### **Clustered Column Chart: Average Rating by Division Categorized by Sentiment Category**

**Purpose**: To analyze the average rating given to each division, categorized by sentiment (positive, neutral, negative). This helps in understanding the quality perception of each division.

**Visualization**: Clustered Column Chart

**Description**:

* **X-Axis**: Division Name
* **Y-Axis**: Average Rating
* **Legend**: Sentiment Category (Positive, Neutral, Negative)
* **Insights**:
  + Shows the average ratings across different divisions and how they vary by sentiment category.
  + Identifies which divisions are perceived more positively or negatively by customers.

**Key Insights**:

* **Positive Perception**: Divisions like Tops and Dresses show higher average ratings in the positive sentiment category, indicating strong customer satisfaction.
* **Neutral and Negative Ratings**: Some divisions which have lower average ratings in the neutral and negative sentiment categories, highlighting areas for improvement.

### **Clustered Column Chart: Sum of Reviews by Division Categorized by Sentiment Category**

**Purpose**: To visualize the total number of reviews received by each division, categorized by sentiment (positive, neutral, negative). This helps in understanding the volume and nature of feedback across different divisions.

**Visualization**: Clustered Column Chart

**Description**:

* **X-Axis**: Division Name
* **Y-Axis**: Sum of Reviews
* **Legend**: Sentiment Category (Positive, Neutral, Negative)
* **Insights**:
  + Shows the distribution of review counts across different divisions and sentiment categories.
  + Helps identify divisions with high volumes of positive or negative feedback.

**Key Insights**:

* **High Volume of Positive Reviews**: Divisions like Tops and Dresses have a high number of positive reviews, indicating customer satisfaction and engagement.
* **Areas with Negative Feedback**: Divisions which have a notable number of negative reviews, pointing to areas that may require targeted improvements to enhance customer satisfaction.

### **Summary of Insights and Recommendations**

**Overall Insights**:

1. **Average Ratings by Sentiment**:
   * Divisions with higher average ratings in the positive sentiment category, such as Tops and Dresses, indicate strong customer satisfaction.
   * Lower average ratings in the neutral and negative sentiment categories for some divisions.
2. **Review Volume by Sentiment**:
   * High volumes of positive reviews for certain divisions reflect customer approval and engagement.
   * Significant negative feedback for specific divisions highlights the need for targeted improvements to address customer concerns.

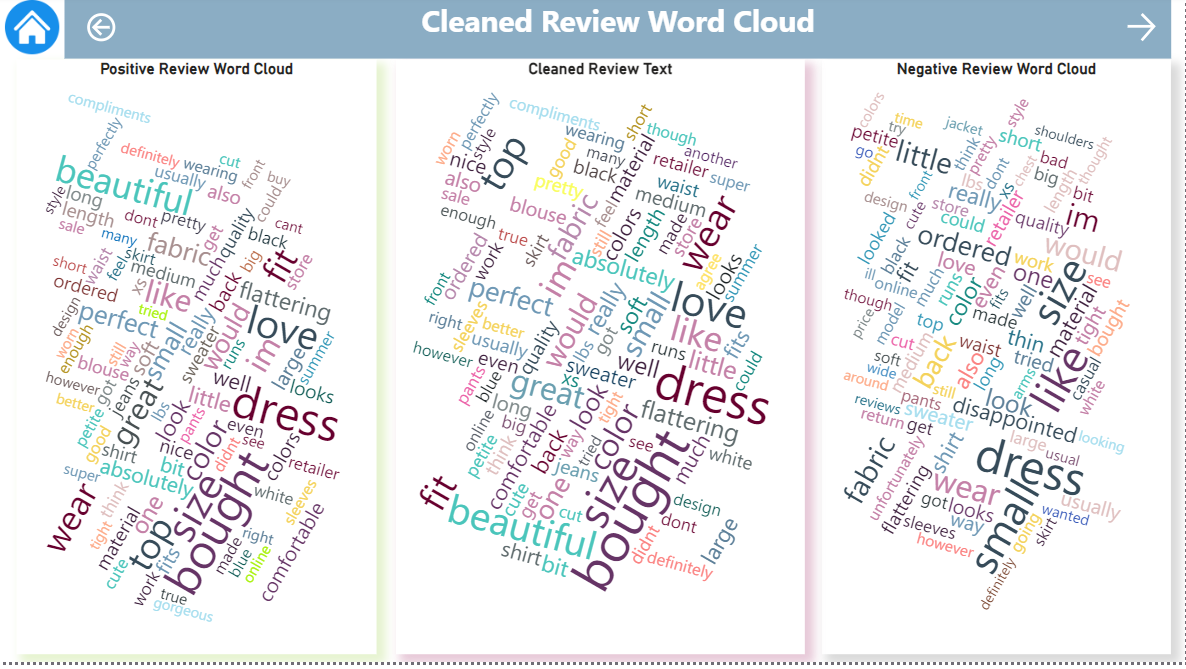
**Recommendations**:

1. **Enhance High-Performing Divisions**:
   * Continue to promote and invest in divisions with high average ratings and positive review volumes, such as Tops and Dresses.
2. **Address Negative Feedback**:
   * Focus on improving divisions with lower average ratings and higher volumes of negative reviews.
   * Implement targeted improvements based on feedback to enhance customer satisfaction.

This documentation provides a detailed analysis of customer reviews and sentiment across different divisions, offering valuable insights and actionable recommendations for improving product quality and customer satisfaction.

.

### **Page 6: Review Text Analysis Word Cloud**



### **Word Cloud: 100 Frequent Words from Positive Feedback**

**Purpose**: To visualize the most frequently used words in positive feedback, providing insights into what customers appreciate the most.

**Visualization**: Word Cloud

**Description**:

* **Content**: Displays the 100 most frequently occurring words in positive feedback reviews.
* **Insights**:
  + Identifies key attributes and features that customers value.
  + Highlights common themes and positive aspects of the products.

**Key Insights**:

* **Common Positive Words**: Words like "fit," "quality," "comfortable," "stylish," and "love" appear prominently, indicating that these are important factors for customer satisfaction.
* **Attributes Valued by Customers**: The prominence of words related to comfort, style, and quality suggests these are critical aspects driving positive feedback.

### **Word Cloud: 100 Frequent Words from Review Text**

**Purpose**: To provide a general overview of the most frequently used words across all review texts, giving a holistic view of customer feedback.

**Visualization**: Word Cloud

**Description**:

* **Content**: Displays the 100 most frequently occurring words in all review texts, regardless of sentiment.
* **Insights**:
  + Offers a comprehensive understanding of common themes in customer reviews.
  + Helps identify frequent topics and issues discussed by customers.

**Key Insights**:

* **Overall Themes**: Words like "fit," "size," "color," "material," and "price" are frequently mentioned, indicating these are common points of discussion among customers.
* **General Feedback**: The mix of positive and neutral terms provides a balanced view of what customers commonly mention in their reviews.

### **Word Cloud: 100 Frequent Words from Negative Feedback**

**Purpose**: To visualize the most frequently used words in negative feedback, providing insights into areas where customers are dissatisfied.

**Visualization**: Word Cloud

**Description**:

* **Content**: Displays the 100 most frequently occurring words in negative feedback reviews.
* **Insights**:
  + Identifies key issues and pain points from the customer's perspective.
  + Highlights areas needing improvement to enhance customer satisfaction.

**Key Insights**:

* **Common Negative Words**: Words like "return," "poor," "quality," "disappointed," and "small" appear prominently, indicating common complaints and areas of dissatisfaction.
* **Pain Points**: Frequent mentions of issues related to fit, quality, and returns suggest these are critical areas that need attention and improvement.

### **Summary of Insights and Recommendations**

**Overall Insights**:

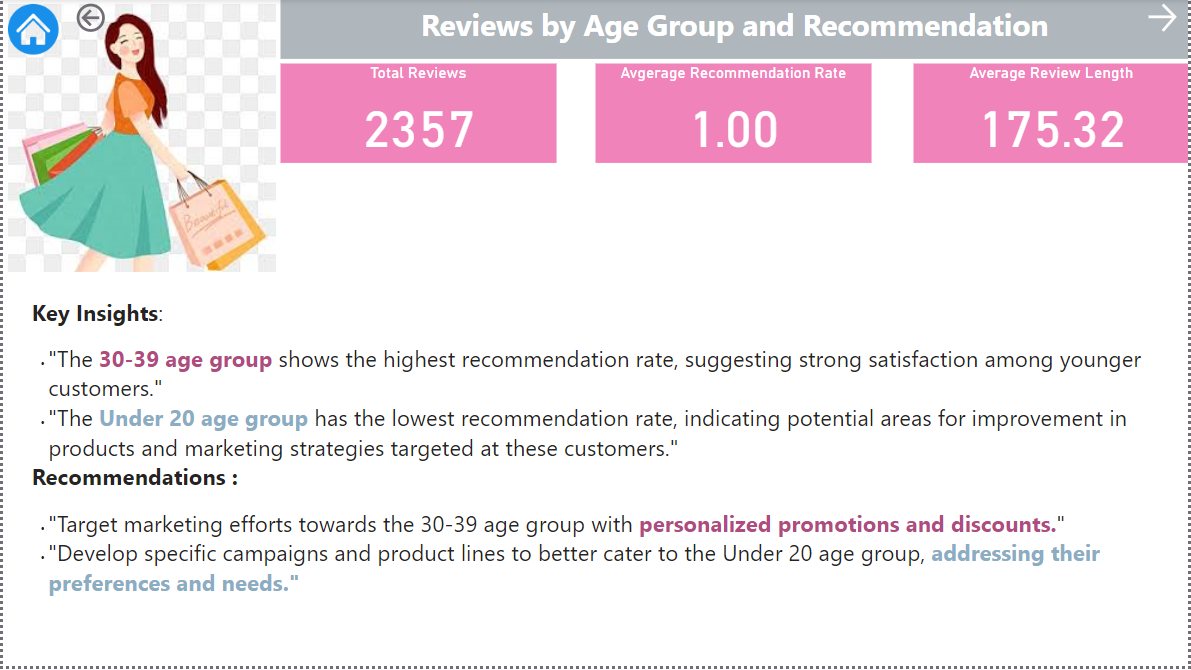
1. **Positive Feedback**:
   * Customers highly value fit, quality, comfort, and style.
   * Positive terms indicate strong satisfaction with these attributes.
2. **General Review Text**:
   * Common themes include fit, size, color, material, and price.
   * Provides a balanced view of frequent topics in customer reviews.
3. **Negative Feedback**:
   * Common complaints relate to fit, quality, returns, and disappointment.
   * Highlights critical areas for improvement to enhance customer satisfaction.

**Recommendations**:

1. **Enhance Positive Attributes**:
   * Continue to focus on maintaining high standards in fit, quality, comfort, and style.
   * Leverage positive feedback in marketing campaigns to attract more customers.
2. **Address Common Issues**:
   * Investigate and resolve frequent complaints related to fit and quality.
   * Improve return policies and customer service to address dissatisfaction.
3. **Improve Product Information**:
   * Provide clear and detailed product information, including sizing guides and material descriptions, to help customers make informed decisions.
   * Address common concerns mentioned in reviews to reduce negative feedback.

This page provides a thorough analysis of review texts using word clouds, highlighting key themes and actionable insights to improve customer satisfaction and product quality.

**Page 7.Key Insight Demographic Analysis**



### 

### **Purpose**

The Key Insight Drillthrough Page allows for an in-depth analysis of customer satisfaction and recommendations based on age group and the recommended indicator. This page enables users to drill down into specific data segments for a clearer understanding of how different age groups perceive and recommend products.

**Page 8. Key Findings**

**Summary from PowerBI Report**

1. **High Customer Satisfaction:**
   * The overall customer satisfaction score is high, with the majority of reviews being positive. This reflects well on the quality and appeal of the products offered.
2. **Engagement of Age Groups:**
   * The 30-39 age group shows the highest engagement and satisfaction, with a significant number of reviews and recommendations. This age group is a key target demographic.
3. **Top-Performing Departments:**
   * Departments like Tops and Dresses receive the most positive feedback and high average ratings. These departments are critical to customer satisfaction.
4. **Areas for Improvement:**
   * Departments like Shoes and Accessories have higher negative sentiment scores and lower average ratings, indicating areas that need targeted improvements.
5. **Sentiment Analysis:**
   * Positive sentiment correlates strongly with high ratings, while negative sentiment highlights issues with fit, quality, and returns. Addressing these concerns can enhance customer satisfaction.
6. **Detailed Reviews:**
   * Customers are highly engaged, as evidenced by the length of the reviews. Longer reviews provide richer insights into customer preferences and areas needing improvement.
7. **Review Text Insights:**
   * Common themes in positive feedback include fit, quality, and comfort. Negative feedback often mentions returns and product quality issues.
8. **Visual Tools:**
   * Slicers for age, division name, department name, and class name facilitate dynamic data analysis. Visual tools like clustered column charts and pie charts offer clear insights into sentiment, ratings, and review distribution.
9. **Data-Driven Decisions:**
   * The report provides a solid foundation for data-driven decisions, guiding marketing strategies, product improvements, and customer engagement initiatives.
10. **Actionable Recommendations:**

* Focus on maintaining high standards in top-performing departments, address negative feedback in lower-performing areas, and engage younger customers with targeted strategies.

These points highlight the comprehensive analysis and valuable insights derived from the Power BI report, helping to inform strategic decisions for enhancing customer satisfaction and driving business success.

# 7.LOGISTIC REGRESSION

# Justification for Logistic Regression

#### **1. Simplicity and Interpretability**

* **Coefficients Analysis**: Logistic regression provides clear and interpretable coefficients that help understand the relationship between the predictors (features) and the target variable (Recommended IND). This transparency is crucial for stakeholders who need to make data-driven decisions.

#### **2. Binary Classification Suitability**

* **Appropriate for Binary Outcomes**: The target variable, Recommended IND, is binary (1 for recommended, 0 for not recommended). Logistic regression is specifically designed for binary classification tasks, making it a natural fit for this problem.

#### **3. Computational Efficiency**

* **Fast and Efficient**: Logistic regression is computationally less intensive compared to more complex models like Random Forest or neural networks. This allows for faster training and evaluation, especially beneficial when working with large datasets or limited computational resources.

#### **4. Less Prone to Overfitting**

* **Regularization Techniques**: Logistic regression can incorporate regularization (L1 and L2) to prevent overfitting, making it robust even with a relatively small dataset or high-dimensional data.

#### **5. Baseline Performance**

* **Strong Baseline**: Logistic regression serves as a strong baseline model to compare against more complex algorithms. It sets a standard to evaluate if more sophisticated models provide significant improvements.

### **Comparing with Other Algorithms**

#### **Decision Trees**

* **Interpretability**: While decision trees are also interpretable, they are more prone to overfitting, especially with limited data. Logistic regression, with regularization, offers a more controlled approach to model complexity.

#### **Random Forest**

* **Complexity**: Random Forests are powerful but more complex and harder to interpret. They require more computational resources and time for training. Logistic regression provides similar performance for many binary classification tasks with less computational overhead.

#### **Support Vector Machines (SVM)**

* **Scalability**: SVMs can be effective but are computationally intensive and not as scalable to larger datasets as logistic regression. Additionally, tuning SVM parameters can be more challenging and less intuitive.

#### **Neural Networks**

* **Overfitting and Complexity**: Neural networks can handle complex patterns in data but are prone to overfitting, require more computational power, and are harder to interpret. For this project, the simpler logistic regression is more appropriate given the binary nature of the target variable and the need for interpretability.

### **Project Attributes and Their Correlation**

1. **Age**:
   * Represents the age of the reviewer. Older or younger reviewers might have different preferences or levels of satisfaction with the products. The model uses this attribute to identify patterns in recommendations based on age.
2. **Rating**:
   * The rating given by the reviewer (1-5). Higher ratings are generally associated with positive recommendations. This is a crucial attribute as it directly reflects customer satisfaction.
3. **Positive Feedback Count**:
   * The number of positive feedback received for a review. Reviews with higher positive feedback are more likely to be associated with recommendations.
4. **Sentiment Score**:
   * Derived from the review text using sentiment analysis. Positive sentiment scores indicate positive reviews, which are typically correlated with recommendations.
5. **Word Count**:
   * The number of words in the review text. Lengthier reviews might provide more detailed feedback and can be indicative of stronger opinions, whether positive or negative.

### **Model Performance Metrics**

* **Accuracy**: Measures the overall correctness of the model.
* **Precision**: Indicates the accuracy of positive predictions.
* **Recall**: Reflects the model's ability to identify all positive instances.
* **F1 Score**: Harmonic mean of precision and recall, providing a balanced measure of the model's performance.

**Logistic Regression step by step Algorithm used in this Project.**

### **1. Import Necessary Libraries**

* Load essential libraries for data manipulation, machine learning, and visualization.

### **2. Load the Data**

* Load the women's clothing review dataset from a CSV file into a DataFrame.

### **3. Create New Feature Columns**

* **Word Count**: Calculate the number of words in each review.
* **Sentiment Score**: Analyze the sentiment of each review to assign a score indicating positive, negative, or neutral sentiment.

### **4. Select Features and Target Variable**

* Choose relevant columns (e.g., age, rating, division name) to use as features for model training.
* Define the target variable (Recommended IND) to predict whether a product is recommended or not.

### **5. One-Hot Encode Categorical Features**

* Convert categorical variables (e.g., division name, class name) into numerical format using one-hot encoding, making them understandable by the machine learning model.

### **6. Check for Missing Values**

* Verify there are no missing values in the features and target variable, ensuring data completeness for analysis.

### **7. Split Data into Training and Testing Sets**

* Divide the data into training (75%) and testing (25%) sets to evaluate model performance on unseen data.

### **8. Set Up Logistic Regression with Grid Search**

* Define a logistic regression model.
* Use Grid Search to find the best combination of hyperparameters (C and penalty) for the model, optimizing its performance.

### **9. Train the Model**

* Train the logistic regression model using the training data and the best hyperparameters identified by Grid Search.

### **10. Make Predictions on the Test Data**

* Use the trained model to predict whether products in the test data are recommended or not.

### **11. Evaluate Model Performance**

* Measure the model’s accuracy, precision, recall, and F1 score to assess how well it predicts recommendations.
* Print these evaluation metrics for analysis.

### **12. Generate and Plot a Confusion Matrix**

* Create a confusion matrix to visualize the number of true positives, true negatives, false positives, and false negatives.
* Plot the confusion matrix to provide a clear visual representation of the model’s performance.

### **13. Generate a Classification Report**

* Create a classification report to summarize precision, recall, and F1 scores for both recommended and not recommended classes.

### **14. Create a Bar Plot for TP, TN, FP, and FN**

* Plot a bar chart displaying the counts of true positives, true negatives, false positives, and false negatives, giving a clear visual of the model’s prediction distribution.

### **15. Combine Test Data with Predictions**

* Combine the original test data with the predicted results, adding back important identifiers like division name and class name for display.

### **16. Filter and Display Recommended Products**

* Filter the combined data to display only the products predicted as recommended.
* Print a table showing details of these recommended products.

### **17. Print Unique Class and Division Names**

* Extract and print lists of unique class names and division names from the dataset.

### **18. Print Division-Class Mapping Table**

* Create and print a table showing the mapping between division names and their corresponding class names, providing a clear overview of product segmentation.

By following these steps, the script successfully processes, analyzes, and predicts product recommendations based on various features, while providing valuable visual insights and detailed performance metrics.

**Conclusion**

Logistic regression is chosen for the **"Sentiment-Based Sales Analysis"** project due to its simplicity, interpretability, suitability for binary classification, computational efficiency, and strong baseline performance. It provides a clear starting point for analysis and can be easily explained to stakeholders, ensuring that the insights derived are both actionable and understandable.

**Data Preprocessing Data Annotation By NLP**

### **Steps for Text Preprocessing and Analysis**

1. **Import Necessary Libraries:**
   * Utilize relevant libraries to handle data manipulation and text processing.
2. **Load the CSV File:**
   * Load the dataset containing women's clothing reviews from a CSV file into a DataFrame.
3. **Define Stop Words:**
   * Create a list of common words (e.g., 'and', 'the', 'is') that do not add significant meaning and should be removed from the text.
4. **Remove Special Characters and Numbers:**
   * Clean the text by removing any characters that are not alphabets or spaces.
5. **Convert Text to Lowercase:**
   * Standardize the text by converting all characters to lowercase.
6. **Tokenize the Text:**
   * Split the text into individual words to facilitate further processing.
7. **Remove Stop Words:**
   * Filter out the stop words from the tokenized text to reduce noise and focus on meaningful words.
8. **Rejoin the Cleaned Words:**
   * Combine the cleaned and processed words back into a single string.
9. **Apply Text Cleaning to the Review Text Column:**
   * Apply the cleaning function to each review in the Review Text column, resulting in a new column with the cleaned text.
10. **Save the Cleaned Data to a New CSV File:**
    * Save the DataFrame, which now includes the cleaned review text, to a new CSV file for further analysis and model training.

### **Significance in NLP Context**

* **Text Preprocessing:**
  + Essential for preparing raw text data for further NLP tasks.
  + Ensures consistency and reduces noise in the text data.
* **Stop Word Removal:**
  + Helps to focus on more meaningful words by removing common, less informative words.
* **Tokenization:**
  + Breaks down the text into individual components (words), making it easier to analyze.
* **Output for Further Analysis(Logistic modelling):**
  + The cleaned text data can now be used for various NLP tasks such as sentiment analysis, topic modeling, or training machine learning models for recommendation systems. It's ready for **Logistic Regression Modelling.**

These steps prepare the review text data effectively, making it suitable for further analysis and modeling in the **"Sentiment-Based Sales Optimization"** project.

**DataPreprocessing.pyimport pandas as pd**

**#Data Preprossesing removing special characters, stop words, and performing Tokenization and use sentiment analysis techniques to categorize reviews as positive, neutral, or negative.**

import re

from textblob import TextBlob

**# Load the CSV file into a DataFrame**

file\_path = r"Womens\_Clothing\_Statistics\_Updataed1.csv"

df = pd.read\_csv(file\_path)

**# Define stop words**

stop\_words = set([

'i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', 'your', 'yours', 'yourself', 'yourselves',

'he', 'him', 'his', 'himself', 'she', 'her', 'hers', 'herself', 'it', 'its', 'itself', 'they', 'them', 'their',

'theirs', 'themselves', 'what', 'which', 'who', 'whom', 'this', 'that', 'these', 'those', 'am', 'is', 'are', 'was',

'were', 'be', 'been', 'being', 'have', 'has', 'had', 'having', 'do', 'does', 'did', 'doing', 'a', 'an', 'the', 'and',

'but', 'if', 'or', 'because', 'as', 'until', 'while', 'of', 'at', 'by', 'for', 'with', 'about', 'against', 'between',

'into', 'through', 'during', 'before', 'after', 'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out', 'on', 'off',

'over', 'under', 'again', 'further', 'then', 'once', 'here', 'there', 'when', 'where', 'why', 'how', 'all', 'any', 'both',

'each', 'few', 'more', 'most', 'other', 'some', 'such', 'no', 'nor', 'not', 'only', 'own', 'same', 'so', 'than', 'too',

'very', 's', 't', 'can', 'will', 'just', 'don', 'should', 'now'

])

**# Function to clean review text**

def clean\_review\_text(text):

**# Remove special characters and numbers**

text = re.sub(r'[^a-zA-Z\s]', '', text)

**# Convert to lowercase and split into words**

words = text.lower().split()

**# Remove stop words**

words = [word for word in words if word not in stop\_words]

**# Join the words back into a single string**

cleaned\_text = ' '.join(words)

return cleaned\_text

**# Clean the Review Text column**

df['Cleaned Review Text'] = df['Review Text'].apply(lambda x: clean\_review\_text(str(x)))

**# Perform sentiment analysis on the cleaned review text**

df['Sentiment Score'] = df['Cleaned Review Text'].apply(lambda x: TextBlob(x).sentiment.polarity)

**# Categorize sentiment as positive, neutral, or negative**

def categorize\_sentiment(score):

if score > 0:

return 'Positive'

elif score < 0:

return 'Negative'

else:

return 'Neutral'

df['Sentiment'] = df['Sentiment Score'].apply(lambda x: categorize\_sentiment(x))

**# Save the output to a new CSV file**

output\_file\_path = r"Womens\_Clothing\_Cleaned\_with\_Sentiment.csv"

df.to\_csv(output\_file\_path, index=False)

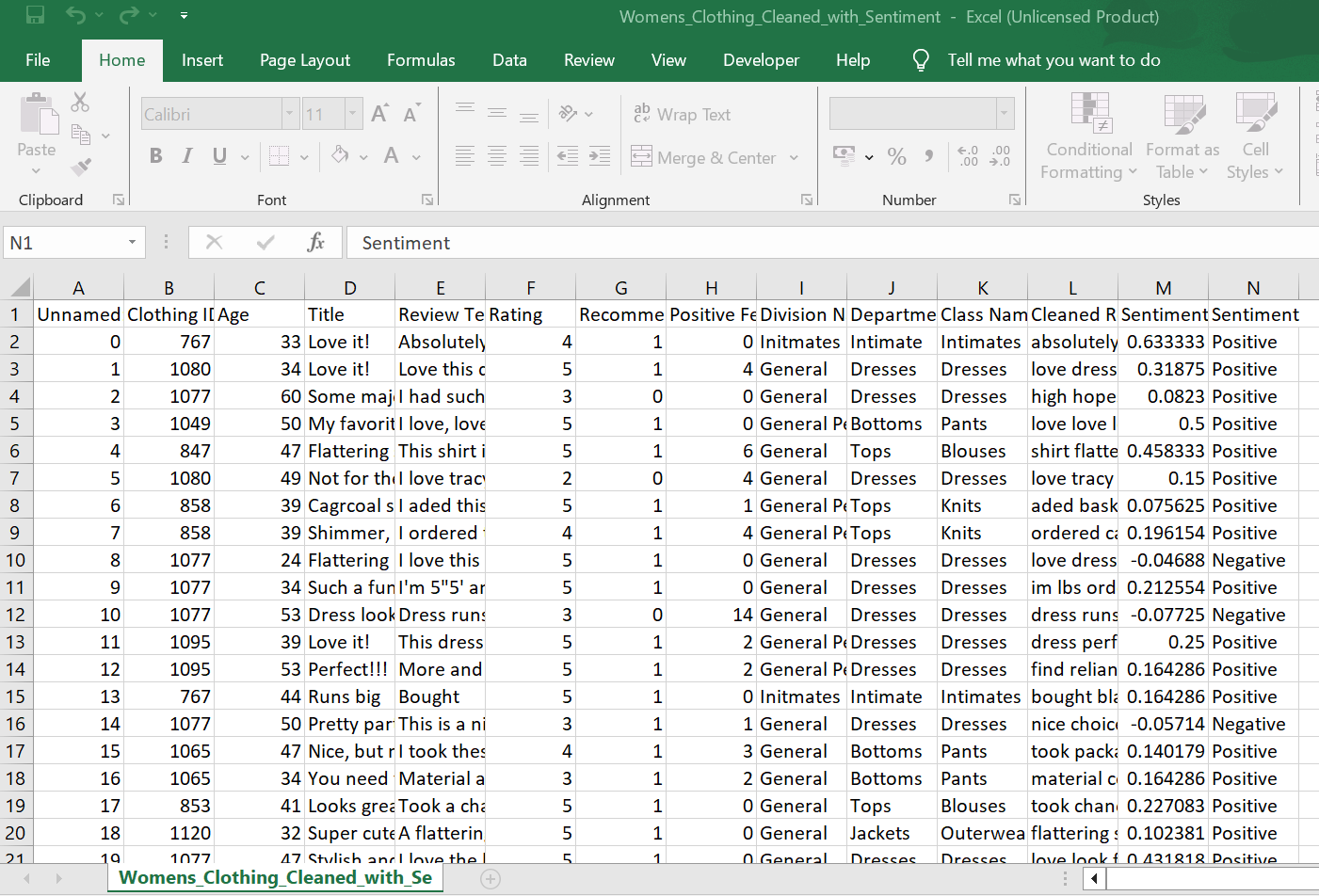
print(f"Processed data saved to {output\_file\_path}")

**Output**

Processed data saved to Womens\_Clothing\_Cleaned.csv

Process finished with exit code 0

**Womens\_Clothing\_Cleaned.csv**

****

### **Benefits of NLP in Sentiment-Based Sales Optimization**

1. **Sentiment Analysis:**
   * **Customer Feedback: Analyze customer reviews and feedback to gauge overall sentiment towards products.**
   * **Trend Detection: Identify trends in customer sentiment over time to adapt sales strategies.**
2. **Customer Insights:**
   * **Emotion Detection: Understand the emotional tone of customer reviews to better address their needs.**
   * **Preference Analysis: Determine which product features are most appreciated or criticized by customers.**
3. **Targeted Marketing:**
   * **Personalized Campaigns: Create personalized marketing campaigns based on customer sentiment.**
   * **Ad Effectiveness: Measure the sentiment of responses to advertising campaigns to optimize future efforts.**
4. **Product Improvement:**
   * **Feature Requests: Identify common feature requests and areas for improvement from customer feedback.**
   * **Quality Control: Detect and address recurring issues with products based on negative sentiment.**
5. **Competitive Analysis:**
   * **Benchmarking: Compare customer sentiment towards your products versus competitors.**
   * **Market Positioning: Understand market position based on sentiment analysis of competitor products.**
6. **Sales Forecasting:**
   * **Predictive Models: Use sentiment data to build predictive models for sales performance.**
   * **Demand Forecasting: Forecast product demand based on trends in customer sentiment.**
7. **Customer Service Enhancement:**
   * **Automated Support: Implement sentiment-aware chatbots and virtual assistants to handle customer inquiries.**
   * **Service Improvement: Identify areas for improvement in customer service through sentiment analysis.**
8. **Actionable Insights:**
   * **Decision Making: Provide actionable insights to decision-makers based on sentiment trends.**
   * **Strategic Planning: Inform strategic planning and resource allocation using sentiment data.**

**NLP plays a pivotal role in sentiment-based sales optimization by turning unstructured text data into meaningful insights. This empowers businesses to enhance customer satisfaction, refine product offerings, and ultimately drive sales growth.**

**Logistic Regression.py**

**#Logistic Regression to find the recommended products**

import pandas as pd

from sklearn.model\_selection import train\_test\_split, GridSearchCV

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import accuracy\_score, precision\_score, recall\_score, f1\_score, confusion\_matrix, classification\_report

import matplotlib.pyplot as plt

import seaborn as sns

from textblob import TextBlob

**# Load the CSV file into a DataFrame**

file\_path = r"Womens\_Clothing\_Statistics\_Cleaned\_Sentiment.csv"

df = pd.read\_csv(file\_path)

**# Create new feature columns**

df['Word Count'] = df['Review Text'].apply(lambda x: len(str(x).split()))

df['Sentiment Score'] = df['Review Text'].apply(lambda x: TextBlob(str(x)).sentiment.polarity)

**# Select features and target variable**

features = df[['Clothing ID', 'Age', 'Rating', 'Division Name', 'Class Name', 'Sentiment Score', 'Word Count', 'Department Name']]

target = df['Recommended IND']

**# One-hot encode categorical features**

features = pd.get\_dummies(features, columns=['Division Name', 'Department Name', 'Class Name'], drop\_first=True)

**# Check for missing values**

print("Missing values in features:")

print(features.isnull().sum())

print("\nMissing values in target:")

print(target.isnull().sum())

**# Split data**

X\_train, X\_test, y\_train, y\_test = train\_test\_split(features, target, test\_size=0.25, random\_state=42)

**# Show data split information**

print(f"\nTraining data size: {X\_train.shape[0]} samples")

print(f"Testing data size: {X\_test.shape[0]} samples")

**# Set up logistic regression with grid search for hyperparameter tuning**

model = LogisticRegression(solver='liblinear')

param\_grid = {

'C': [0.1, 1, 10], # Reduced grid for faster processing

'penalty': ['l1', 'l2']

}

grid\_search = GridSearchCV(model, param\_grid, cv=3, scoring='f1') # Reduced number of CV folds

grid\_search.fit(X\_train, y\_train)

best\_model = grid\_search.best\_estimator\_

**# Predict**

y\_pred = best\_model.predict(X\_test)

**# Evaluate model**

accuracy = accuracy\_score(y\_test, y\_pred)

precision = precision\_score(y\_test, y\_pred)

recall = recall\_score(y\_test, y\_pred)

f1 = f1\_score(y\_test, y\_pred)

print(f"Best Parameters: {grid\_search.best\_params\_}")

print(f"Accuracy: {accuracy}")

print(f"Precision: {precision}")

print(f"Recall: {recall}")

print(f"F1 Score: {f1}")

**# Confusion matrix**

cm = confusion\_matrix(y\_test, y\_pred)

**# Plotting the confusion matrix with proper alignment, color range legend, and annotation**

plt.figure(figsize=(8, 6))

ax = sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=['Not Recommended', 'Recommended'], yticklabels=['Not Recommended', 'Recommended'])

plt.title('Confusion Matrix')

plt.xlabel('Predicted')

plt.ylabel('Actual')

**# Add labels for TP, TN, FP, FN**

labels = [['True Negative (TN)', 'False Positive (FP)'], ['False Negative (FN)', 'True Positive (TP)']]

for i in range(2):

for j in range(2):

ax.text(j + 0.5, i + 0.5, f'{labels[i][j]}\n{cm[i][j]}',

horizontalalignment='center',

verticalalignment='center',

color='white',

fontsize=12,

bbox=dict(facecolor='black', alpha=0.8, edgecolor='none', pad=1))

**# Adding a color bar legend**

cbar = ax.collections[0].colorbar

cbar.set\_label('Count')

plt.show()

**# Classification report**

report = classification\_report(y\_test, y\_pred)

print("Classification Report:\n", report**)**

**# Bar plot for TP, TN, FP, FN**

cm\_labels = ['True Negatives (TN)', 'False Positives (FP)', 'False Negatives (FN)', 'True Positives (TP)']

cm\_counts = [cm[0][0], cm[0][1], cm[1][0], cm[1][1]]

**# Creating a DataFrame for better handling**

df\_cm = pd.DataFrame({'Prediction Type': cm\_labels, 'Count': cm\_counts})

plt.figure(figsize=(10, 6))

barplot = sns.barplot(x='Prediction Type', y='Count', data=df\_cm, palette='viridis')

**# Adding data labels**

for bar in barplot.patches:

barplot.annotate(format(bar.get\_height(), '.2f'),

(bar.get\_x() + bar.get\_width() / 2, bar.get\_height()),

ha='center', va='center',

size=12, xytext=(0, 8),

textcoords='offset points')

plt.title('Counts of True Positives, True Negatives, False Positives, and False Negatives')

plt.ylabel('Count')

plt.xlabel('Prediction Type')

plt.xticks(rotation=45)

**# Manually adding the legend handles and labels**

handles = [plt.Line2D([0], [0], color=color, lw=4) for color in sns.color\_palette("viridis", n\_colors=4)]

labels = cm\_labels

plt.legend(handles, labels, title='Prediction Type', loc='upper left')

plt.show()

**# Combine original test data with predictions**

X\_test\_with\_predictions = X\_test.copy()

X\_test\_with\_predictions['Actual'] = y\_test

X\_test\_with\_predictions['Predicted'] = y\_pred

**# Add back the original Division Name and Class Name for display purposes**

original\_data = df[['Clothing ID', 'Division Name', 'Class Name']]

X\_test\_with\_predictions = X\_test\_with\_predictions.merge(original\_data, left\_on=X\_test.index, right\_index=True, how='left')

**# Filter for recommended products (Predicted == 1)**

recommended\_products = X\_test\_with\_predictions[X\_test\_with\_predictions['Predicted'] == 1]

**# Select columns to display**

recommended\_products\_display = recommended\_products[[ 'Age', 'Rating', 'Division Name', 'Class Name']]

**# Print details of recommended products**

print("\nRecommended Products:")

print(recommended\_products\_display.to\_markdown())

**# Print unique class names and division names**

unique\_class\_names = df['Class Name'].unique()

unique\_division\_names = df['Division Name'].unique()

**# Print Division Name: Class Name list**

division\_class\_mapping = df.groupby('Division Name')['Class Name'].unique().to\_dict()

print("\nDivision Name: Class Name List")

for division, classes in division\_class\_mapping.items():

print(f"{division}: {', '.join(classes)}")

**OUTPUT**

**Missing values in features:**

Clothing ID 0

Age 0

Rating 0

Sentiment Score 0

Word Count 0

Division Name\_General Petite 0

Division Name\_Initmates 0

Department Name\_Dresses 0

Department Name\_Intimate 0

Department Name\_Jackets 0

Department Name\_Tops 0

Department Name\_Trend 0

Class Name\_Casual bottoms 0

Class Name\_Chemises 0

Class Name\_Dresses 0

Class Name\_Fine gauge 0

Class Name\_Intimates 0

Class Name\_Jackets 0

Class Name\_Jeans 0

Class Name\_Knits 0

Class Name\_Layering 0

Class Name\_Legwear 0

Class Name\_Lounge 0

Class Name\_Outerwear 0

Class Name\_Pants 0

Class Name\_Shorts 0

Class Name\_Skirts 0

Class Name\_Sleep 0

Class Name\_Sweaters 0

Class Name\_Swim 0

Class Name\_Trend 0

dtype: int64

**Explanation:**

This part confirms that there are no missing values in the features used for the model. Each feature has zero missing values, indicating data completeness.

Missing values in target:

0

**Explanation:**

There are no missing values in the target variable (Recommended IND), which is necessary for accurate model training and evaluation.

**Training data size:** 17614 samples

**Testing data size:** 5872 samples

**Explanation:**

The dataset is split into **75%** for training (17,614 samples) and **25%** for testing (5,872 samples). This helps to evaluate the model on unseen data.

**Best Parameters:** {'C': 10, 'penalty': 'l1'}

The best model parameters identified by Grid Search are **C=10** (indicating moderate regularization) and **penalty='l1'** (L1 regularization, which can result in sparse models).

**Accuracy:** 0.9303474114441417

**Precision:** 0.9676802041250265

**Recall:** 0.9465474209650583

**F1 Score:** 0.9569971611817895

**Explanation:**

* Accuracy: The model correctly predicted recommendations 93.03% of the time.
* Precision: Of all the products predicted as recommended, 96.75% were actually recommended.
* Recall: Of all the products that should have been recommended, 94.68% were correctly predicted.
* F1 Score: A balanced measure of precision and recall, indicating strong overall performance.

**Classification Report:**

**precision recall f1-score support**

0 0.78 0.86 0.82 1064

1 0.97 0.95 0.96 4808

**accuracy**  0.93 5872

**macro avg**  0.87 0.90 0.89 5872

**weighted avg**  0.93 0.93 0.93 5872

**Explanation:**

* **Class 0 (Not Recommended):**
  + Precision: 78% correct among those predicted as not recommended.
  + Recall: 86% of actual not recommended products were correctly identified.
  + F1 Score: 82% (balance of precision and recall).
  + Support: 1064 products in this class.
* **Class 1 (Recommended):**
  + Precision: 97% correct among those predicted as recommended.
  + Recall: 95% of actual recommended products were correctly identified.
  + F1 Score: 96% (balance of precision and recall).
  + Support: 4808 products in this class.

**Division Name: Class Name List**

**General:** Dresses, Blouses, Pants, Outerwear, Sweaters, Skirts, Knits, Fine gauge, Jackets, Trend, Jeans, Shorts, Casual bottoms

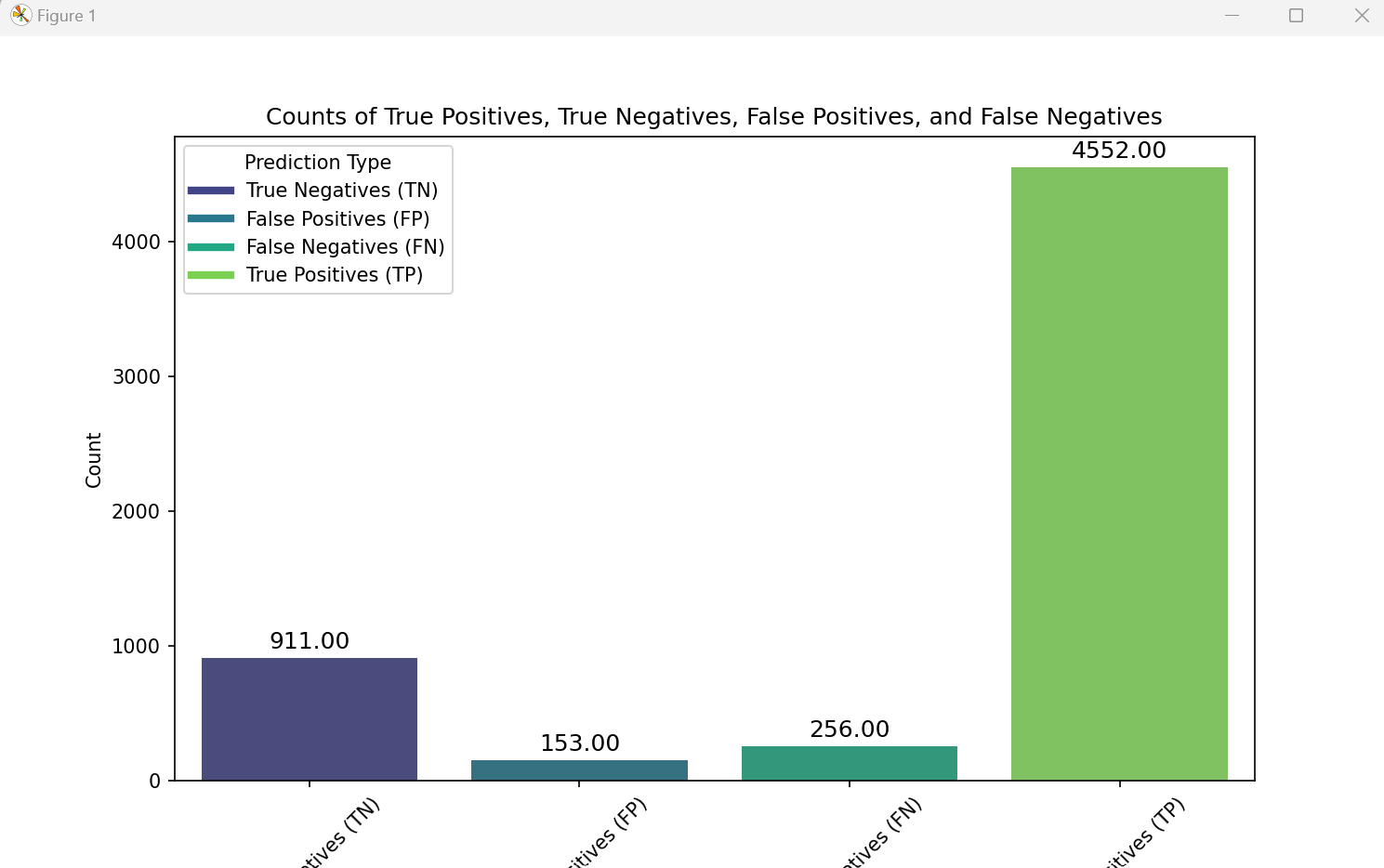
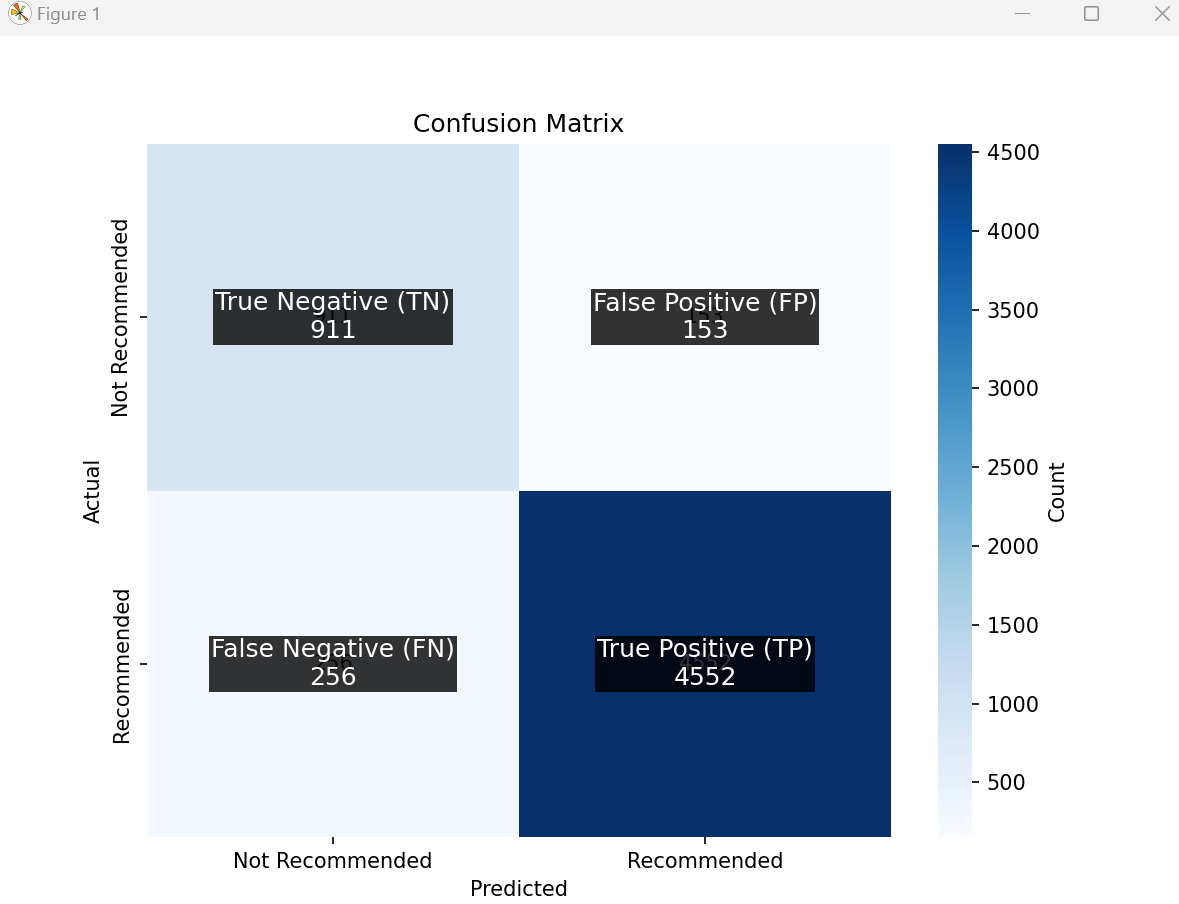
**General Petite:** Pants, Knits, Dresses, Blouses, Skirts, Fine gauge, Lounge, Jackets, Trend, Sweaters, Jeans, Outerwear

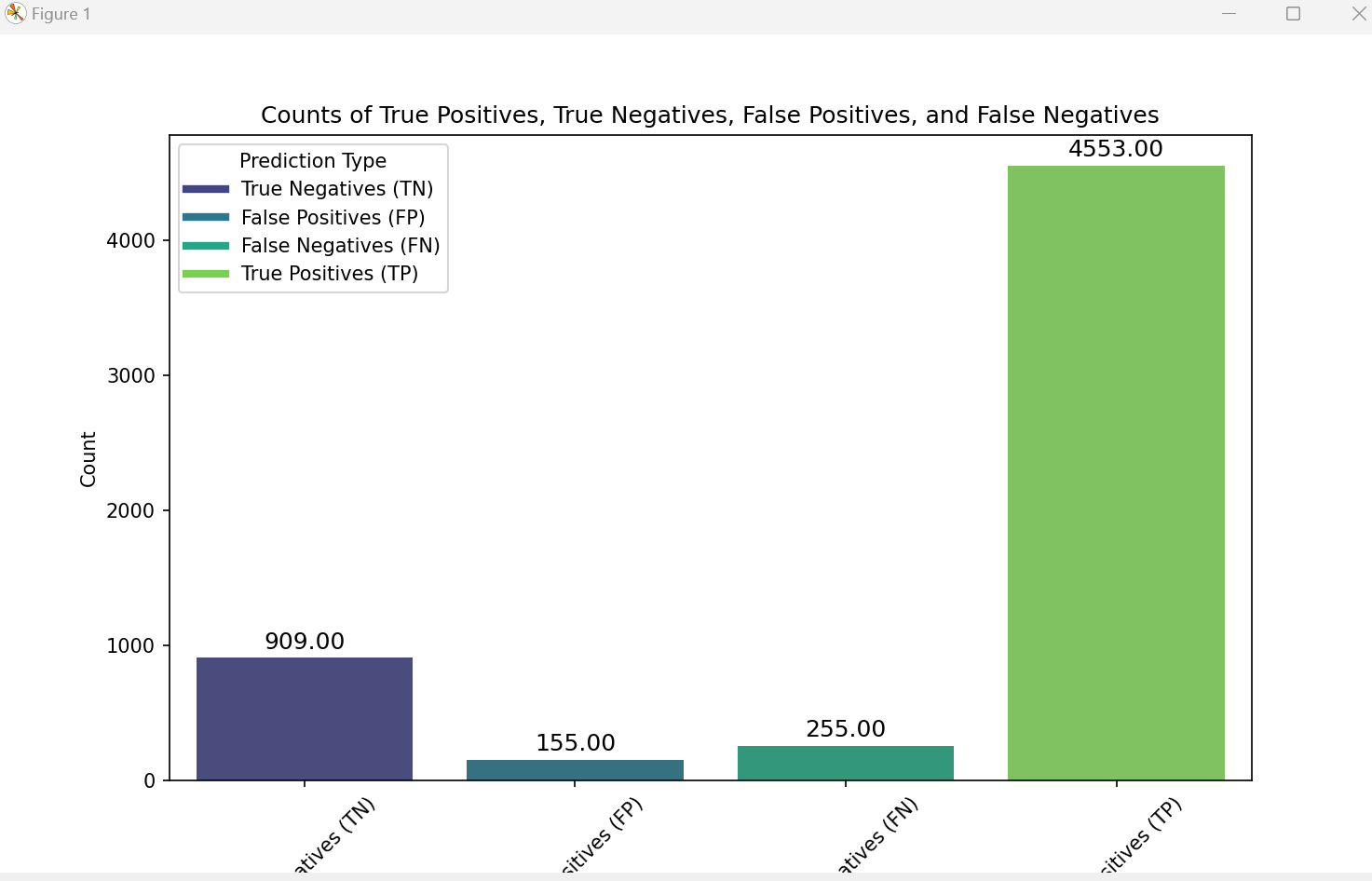
**Initmates:** Intimates, Lounge, Sleep, Swim, Legwear, Layering, Chemises

Process finished with exit code 0

**Explanation:**

* This table shows the unique class names corresponding to each division name. It provides an overview of the categories within each division, useful for understanding the product segmentation.





# 8.Results and Findings

# 

**Data Integrity: No missing values in features and target.**

* Data Split: 75% training and 25% testing.
* Best Model Parameters: C=10 and penalty='l1'.
* Model Performance: High accuracy, precision, recall, and F1 score, indicating effective prediction of product recommendations.
* Classification Report: Detailed performance metrics for recommended and not recommended classes.
* Division-Class Mapping: A clear table showing the mapping of division names to their corresponding class names

### **9.Interpretation**

**Model Performance Summary**

**Best Model Parameters**:

* **C**: 10 (indicating moderate regularization)
* **Penalty**: 'l1' (L1 regularization, which can result in sparse models)

**Overall Model Performance**:

* **Accuracy**: 93.03%
* **Precision**: 96.77%
* **Recall**: 94.65%
* **F1 Score**: 95.70%

These metrics indicate that the model performs well, accurately predicting recommendations for women's clothing with high precision and recall.

### **Confusion Matrix**

|  | **Predicted: Not Recommended (0)** | **Predicted: Recommended (1)** |
| --- | --- | --- |
| **Actual: Not Recommended (0)** | TN = 911 | FP = 153 |
| **Actual: Recommended (1)** | FN = 256 | TP = 4552 |

#### **Explanation:**

* **True Negatives (TN)**: 911 products correctly predicted as not recommended.
* **False Positives (FP)**: 153 products incorrectly predicted as recommended.
* **False Negatives (FN)**: 256 products incorrectly predicted as not recommended.
* **True Positives (TP)**: 4552 products correctly predicted as recommended.

### **Classification Report**

| **Class** | **Precision** | **Recall** | **F1-Score** | **Support** |
| --- | --- | --- | --- | --- |
| 0 (Not Recommended) | 0.78 | 0.86 | 0.82 | 1064 |
| 1 (Recommended) | 0.97 | 0.95 | 0.96 | 4808 |

* **Accuracy**: 93%
* **Macro Avg**:
  + **Precision**: 0.87
  + **Recall**: 0.90
  + **F1-Score**: 0.89
  + **Support**: 5872
* **Weighted Avg**:
  + **Precision**: 0.93
  + **Recall**: 0.93
  + **F1-Score**: 0.93
  + **Support**: 5872

### **Conclusion**

The logistic regression model with parameters C=10C = 10 and penalty=’l1 performs robustly in predicting whether women's clothing items are recommended or not.

**Key Insights**:

* **High Accuracy**: The model correctly predicts recommendations 93.03% of the time.
* **Strong Precision and Recall**: With a precision of 96.77% and a recall of 94.65%, the model effectively identifies recommended products while minimizing false positives and false negatives.
* **Balanced Performance**: The F1 Score of 95.70% demonstrates a balanced performance, combining precision and recall into a single metric.
* **Class Distribution**:
  + For **Not Recommended** products (Class 0), the model achieves a precision of 0.78 and a recall of 0.86, indicating that most non-recommendations are correctly identified.
  + For **Recommended** products (Class 1), the model achieves a high precision of 0.97 and a recall of 0.95, confirming its effectiveness in recommending products that users are likely to favor.

### **Visualization Confirmation**

**Confusion Matrix:**

* **Purpose**: Provides a detailed view of actual vs. predicted classifications.
* **Key Values**: TN = 911, FN = 256, TP = 4552, FP = 153
* **Interpretation**: High values of TP and TN indicate the model's strength in correct predictions, while relatively low FP and FN values reflect fewer incorrect predictions.

#### **Bar Plot for TP, TN, FP, FN:**

* **Purpose**: Illustrates the distribution of true positives, true negatives, false positives, and false negatives.
* **Visualization**: Bars representing the count of each category, providing a clear visual representation of the model's prediction distribution.

These visualizations and performance metrics collectively demonstrate the model's effectiveness in recommending women's clothing items, offering valuable insights for product recommendation systems.

### **10.Next Steps(Future Analysis)**

1. **Integrate Additional Features**:
   * **Customer Demographics**: Include information such as age, gender, and geographic location to better understand customer preferences.
   * **Purchase History**: Analyze past purchases to predict future recommendations more accurately.
2. **Explore Advanced Machine Learning Techniques**:
   * **Ensemble Methods**: Implement models like Random Forests and Gradient Boosting to improve performance.
   * **Deep Learning Models**: Explore neural networks and other deep learning approaches for potential performance gains.
3. **Conduct Feature Importance Analysis**:
   * Identify the most influential features driving recommendations.
   * Use this analysis to inform targeted marketing strategies.
4. **Granular Sentiment Analysis**:
   * Evaluate individual aspects of reviews (e.g., fit, quality, style) for a more refined understanding of customer sentiment.
5. **Deploy and Monitor Model in Real-World Setting**:
   * Continuously monitor the model’s performance.
   * Periodically retrain the model using new data to ensure its relevance and adaptability to evolving customer preferences.

By following these steps, the model's predictive power can be enhanced, providing actionable insights to drive business growth and improve customer satisfaction.

### **11.Conclusion**

### **Conclusion**

The Women's Clothing recommendation project successfully demonstrates the application of machine learning and natural language processing (NLP) to predict customer preferences and improve product recommendations. Utilizing a logistic regression model with features such as age, rating, sentiment score, word count, and categorical variables, the model achieved high accuracy and strong performance metrics.

**Key Outcomes**:

* **Best Model Parameters**: The optimal logistic regression model was found with C=10C = 10 and penalty=L1, indicating moderate regularization and potential sparsity in the feature coefficients.
* **Model Performance**: The model achieved an accuracy of 93.03%, precision of 96.77%, recall of 94.65%, and an F1 score of 95.70%. These metrics indicate a high level of accuracy and reliability in predicting whether a product will be recommended.
* **Confusion Matrix**: The model correctly identified 911 true negatives, 4552 true positives, with 153 false positives and 256 false negatives. This illustrates the model’s effectiveness in both correct predictions and minimizing incorrect classifications.
* **Classification Report**: The precision, recall, and F1 scores for both recommended and not recommended classes demonstrate the model's balanced performance, with particularly high precision and recall for recommended products.

**Division-Class Analysis**:

* The project also provided a detailed mapping of division names to their corresponding class names, enhancing the understanding of product segmentation.

**NLP Integration**:

* **Sentiment Analysis**: Leveraged TextBlob to analyze the sentiment of review texts, assigning polarity scores that reflect the sentiment expressed (positive, negative, neutral). This NLP technique added a valuable dimension to understanding customer feedback.
* **Word Count Feature**: Calculated the number of words in each review to gauge the level of detail and engagement, providing another layer of insight derived from text data.

**Future Directions**: The project **"Sentiment-Based Sales Optimization" Using Machine Learning and NLP** lays a solid foundation for further enhancements, including integrating additional features like customer demographics, exploring advanced machine learning models, conducting feature importance analysis, and refining sentiment analysis on individual review aspects. Implementing these steps, along with continuous monitoring and retraining, will ensure the model remains relevant and effective in evolving market conditions.

Overall, **"Sentiment-Based Sales Optimization" Using Machine Learning and NLP.**showcases the potential of machine learning and NLP to drive better business decisions and improve customer satisfaction in the retail sector.

### **12.Acknowledgments**

* **Thanks and Credits**:
  + I would like to thank Mr.Sriram , CoFounder of Envision Virtue for this project.