**Analysis of supply chin data**

**Data Cleaning and Transformation**

Before proceeding with analysis, I conducted an initial assessment of the dataset to identify any potential issues, such as missing values, duplicates, and data inconsistencies. Below are the steps I took, along with the Python functions used for each task.

**1. Data Overview:**

To understand the structure of the dataset, I used the info() function, which provided a summary of the dataset including:

* Number of rows and columns
* Data types of each column (e.g., integers, floats, objects)
* A screenshot of a computer screen

  Description automatically generatedThe number of non-null values in each column

**2. Checking for Null or Empty Values:**

To ensure the dataset was complete, I created a function to check for any missing or null values across all columns. Detecting null values early is critical for avoiding issues later in the analysis, such as inaccurate computations or visualizations.

**Result:** there’s not any messing value in the data.

**3.Detecting Duplicate Records:**

Duplicate records can distort analysis results, so I wrote a function to check for duplicates. This ensured that each row in the dataset was unique and represented a distinct data point.

**Result**: Total duplicate rows: 0

**Data Enhancements: New Columns Added**

To enhance the dataset and derive deeper insights, I created three additional columns that provide more comprehensive financial data. These columns give us a better understanding of the relationship between costs, profits, and operational efficiency. Below is a breakdown of each new column and how they were calculated.

**1. Total Cost:**

This column represents the sum of **Manufacturing Costs** and **Operational Costs**. It gives us the complete operational expenditure associated with each product, which includes both the production and shipping costs.

Formula used:

python

Copy code

df['Total Cost'] = df['Manufacturing costs'] + df['Costs']

**Rationale**: Understanding the total cost of each product is crucial for evaluating profitability and comparing it with the generated revenue.

**2. Operational Profit:**

The **Operational Profit** column represents the difference between **Revenue Generated** and **Total Cost**. This value indicates how much profit remains after covering both manufacturing and shipping costs.

Formula used:

python

df['Operational Profit'] = df['Revenue generated'] - df['Total Cost']

**Rationale**: Operational profit gives a clear picture of the profitability of each product, factoring in all associated costs.

**3. Profit Margin:**

The **Profit Margin** column was added to provide a percentage-based measure of profitability. It is calculated as the ratio of operational profit to revenue, giving an insight into how efficiently each product generates profit relative to its revenue.

Formula used:

python

df['Profit Margin'] = (df['Operational Profit'] / df['Revenue generated']) \* 100

**Rationale**: The profit margin helps us understand the relative profitability of each product, offering a more standardized way to compare products of different sizes and revenues.

Including these columns in the dataset allows for a more detailed analysis of cost and profit distribution across products. It helps stakeholders focus on the most profitable products, identify cost inefficiencies, and make data-driven decisions regarding pricing, marketing, and supplier relationships.