Name: Nitish Kumar

Reg No. : 12209647

Project: Inventory Management System

# Questions

Create an inventory management system that tracks and analyzes stock levels, sales, and orders. This system will use Numpy for numerical computations, Pandas for data manipulation, and Matplotlib/Seaborn for data visualization.

Import the libraries and load the datasets.

"""

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

df = pd.read\_csv("/content/inventory\_management\_System\_dataset.csv")

df

"""# Data Overview and Analysis

Print the 5 rows of starting.

"""

df.head()

"""Prints the 5 rows of end"""

df.tail()

"""Its provides the descriptive statistics for the datasets"""

df.describe()

"""to show the datatypes of each columns"""

df.info()

"""no of rows and columns"""

df.shape

"""to show the null values of each columns"""

df.isnull()

"""To show the number of null values of each columns"""

df.isnull().sum()

"""# Analyzing Stock Levels

Calculate Total Stock Quantity per Product

"""

total\_stock = df.groupby('ProductName')['TotalItemQuantity'].sum().reset\_index()

total\_stock

"""plot total stock quantity per products"""

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

sns.barplot(data=total\_stock, x='ProductName', y='TotalItemQuantity', palette='viridis')

plt.title("Total stock quantity per Product")

plt.xticks(rotation=45)

plt.show()

"""# Analyzing Sales..

# total sales Amount per Product.

"""

total\_sales = df.groupby('ProductName')['SalesAmount'].sum().reset\_index()

total\_sales

"""# Plot total sales amount per product

"""

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

sns.barplot(data=total\_sales, x='ProductName', y='SalesAmount', palette='viridis')

plt.title('Total Sales Amount per Product')

plt.xticks(rotation=45)

plt.show()

df

"""# Monthly Sales Analysis"""

# extract the month and year from orderdate

df['OrderDate'] = pd.to\_datetime(df['OrderDate'])

df['Month'] = df['OrderDate'].dt.to\_period('M')

"""# Calculate total sales amount per month

"""

monthly\_sales = df.groupby('Month')['SalesAmount'].sum().reset\_index()

"""# plot monthly sales Amount"""

"""# Analyzing Orders

# plot order status Distribution

"""

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

sns.countplot(data=df, x='Status', hue = 'SalesOrderStatus', palette='viridis')

plt.title('Order Status Distribution')

plt.show()

"""# Sales Order Analysis

# Calculate total sales amount per order status

"""

sales\_order\_status = df.groupby('SalesOrderStatus')['SalesAmount'].sum().reset\_index()

sales\_order\_status

"""# total sales amount per order status"""

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

sns.barplot(data=sales\_order\_status, x='SalesOrderStatus', y='SalesAmount',hue = 'SalesOrderStatus', palette='viridis', dodge=False)

plt.title('Total Sales Amount per Order Status')

plt.show()

"""# Advanced Analysis with Numpy

Average profit per product.

"""

average\_profit = df.groupby('CategoryName')['Profit'].mean().reset\_index()

""" Plot average profit per product category

"""

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

sns.barplot(data=average\_profit, x='CategoryName', y='Profit', palette='viridis')

plt.title('Average Profit per Product Category')

plt.show()

"""# Insights and Reporting

# Summary Statistics

summary statistics of sales amount

"""

sales\_summary = df['SalesAmount'].describe()

sales\_summary

"""summary statistics of product standard cost

"""

cost\_summary = df['ProductStandardCost'].describe()

print(cost\_summary)

"""# Correlation Analysis

correaltion Analysis

"""

correlation\_matrix = df[['SalesAmount', 'ProductStandardCost', 'Profit', 'OrderItemQuantity', 'TotalItemQuantity']].corr()

correlation\_matrix

"""

# plot heatmap of the correlation matrix"""

plt.figure(figsize=(10, 8))

sns.heatmap(correlation\_matrix, annot=True, cmap='viridis')

plt.title('Correlation Matrix')

plt.show()