1. Problem Description

The dataset contains sales transaction details on Amazon, including order ID, date, status, fulfillment method, sales channel, product category, size, quantity, amount, and shipping details.

The goal is to analyze this data to derive actionable insights that support business decision-making.

2. Key Objectives

* Sales Overview: Analyze sales performance, trends, and patterns over time.
* Product Analysis: Evaluate product categories, sizes, and quantities sold to identify popular products.
* Fulfillment Analysis: Assess the effectiveness of fulfillment methods.
* Customer Segmentation: Group customers based on buying behavior, location, and other relevant factors.
* Geographical Analysis: Explore sales distribution across states and cities.
* Business Insights: Provide recommendations for improving sales strategies and customer satisfaction.

3. Deliverables

* Comprehensive analysis report summarizing findings, insights, and recommendations.
* Visualizations (charts, graphs) illustrating key data trends.
* Insights on product preferences, customer behavior, and geographical sales.
* Recommendations for improving sales strategies, inventory management, and customer service.

4. Expected Outcome

* Gaining insights to optimize business operations and drive revenue growth.
* Providing actionable recommendations tailored to the business's specific challenges and goals.This primary goal is to leverage insights from the dataset to:
* Optimize business operations by identifying inefficiencies.
* Enhance customer experience through targeted strategies and improved fulfillment.
* Drive revenue growth by focusing on popular products and high-performing regions.
* Develop actionable strategies that align with business goals and challenges.

This analysis empowers the business with actionable recommendations tailored to its unique challenges, providing the foundation for continuous improvement and long-term success

# Import required libraries

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

# Load the dataset

file\_path = "C:\Users\HP\Downloads\Amazon Sale Report (1).csv" # Replace with your file path

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

# Display basic info about the dataset

print("Dataset Info:")

print(data.info())

# ====== Data Cleaning ======

# 1. Remove duplicate rows

data = data.drop\_duplicates()

print(f"Number of duplicates removed: {data.duplicated().sum()}")

# 2. Handle missing values

missing\_summary = data.isnull().sum()

print("\nMissing Values Summary:")

print(missing\_summary)

# Impute missing numerical columns with mean

num\_cols = data.select\_dtypes(include=['float64', 'int64']).columns

for col in num\_cols:

data[col].fillna(data[col].mean(), inplace=True)

# Impute missing categorical columns with mode

cat\_cols = data.select\_dtypes(include=['object']).columns

for col in cat\_cols:

data[col].fillna(data[col].mode()[0], inplace=True)

# Drop columns with all missing values (if any)

data = data.dropna(axis=1, how='all')

# 3. Outlier Handling (Capping)

# Use the IQR method to detect and cap outliers in numerical columns

def cap\_outliers(df, col):

Q1 = df[col].quantile(0.25)

Q3 = df[col].quantile(0.75)

IQR = Q3 - Q1

lower\_bound = Q1 - 1.5 \* IQR

upper\_bound = Q3 + 1.5 \* IQR

df[col] = np.where(df[col] < lower\_bound, lower\_bound, df[col])

df[col] = np.where(df[col] > upper\_bound, upper\_bound, df[col])

for col in ['Amount', 'Qty']: # Adjust based on your data

cap\_outliers(data, col)

# ====== Univariate Analysis ======

print("\nNumerical Summary:")

print(data.describe())

# Distribution of 'Amount'

plt.figure(figsize=(8, 5))

sns.histplot(data['Amount'], bins=50, kde=True, color='blue')

plt.title('Distribution of Amount')

plt.show()

# Distribution of 'Qty'

plt.figure(figsize=(8, 5))

sns.histplot(data['Qty'], bins=30, kde=True, color='green')

plt.title('Distribution of Quantity')

plt.show()

# Count plot for 'Category'

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

sns.countplot(data=data, x='Category', order=data['Category'].value\_counts().index, palette='Set2')

plt.title('Count of Orders by Category')

plt.xticks(rotation=45)

plt.show()

# ====== Bivariate Analysis ======

# Correlation heatmap

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

correlation = data.corr()

sns.heatmap(correlation, annot=True, cmap='coolwarm', fmt='.2f')

plt.title('Correlation Heatmap')

plt.show()

# Scatter plot: 'Amount' vs 'Qty'

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

sns.scatterplot(data=data, x='Qty', y='Amount', hue='Category', palette='Set1')

plt.title('Scatter Plot: Quantity vs Amount')

plt.show()

# Box plot: Amount by Category

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

sns.boxplot(data=data, x='Category', y='Amount', palette='Set3')

plt.title('Box Plot of Amount by Category')

plt.xticks(rotation=45)

plt.show()

# Bar plot: Average Amount by Sales Channel

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

avg\_amount\_by\_channel = data.groupby('Sales Channel')['Amount'].mean().sort\_values()

avg\_amount\_by\_channel.plot(kind='bar', color='skyblue')

plt.title('Average Amount by Sales Channel')

plt.ylabel('Average Amount')

plt.xlabel('Sales Channel')

plt.show()

# Geographical analysis (if columns like 'ship-country' or 'ship-city' exist)

if 'ship-country' in data.columns:

country\_sales = data.groupby('ship-country')['Amount'].sum().sort\_values(ascending=False)

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

country\_sales[:10].plot(kind='bar', color='purple')

plt.title('Top 10 Countries by Total Sales')

plt.ylabel('Total Sales')

plt.xlabel('Country')

plt.show()

# Pairplot for numerical features

sns.pairplot(data[['Amount', 'Qty']], diag\_kind='kde', corner=True)

plt.suptitle('Pairplot of Numerical Features', y=1.02)

plt.show()

# ====== Insights ======

# Print key insights

print("\nKey Insights:")

print("1. The most popular product category is:", data['Category'].mode()[0])

print("2. The average order amount is:", round(data['Amount'].mean(), 2))

print("3. The maximum quantity sold in a single transaction is:", data['Qty'].max())

# Save the cleaned data

data.to\_csv('/mnt/data/cleaned\_amazon\_sales.csv', index=False)

print("\nCleaned dataset saved as 'cleaned\_amazon\_sales.csv'.")