



Amazon Sales Report Analysis

Kevin Amala Aakash A

Kevin

+91 9080183349

Kevin.a.aakash@gmail.com

Project link : <https://github.com/AgentVIPER-k>

InnoByte Amazon Sales Report Analysis Project

Project Overview

The objective of this project is to conduct a comprehensive analysis of an Amazon sales dataset to extract actionable insights that can support business decision-making. The dataset contains details about sales transactions, including order ID, date, status, fulfilment method, sales channel, product category, size, quantity, amount, shipping details, and more.

Key Objectives

1. Sales Overview: Understand overall sales performance, trends, and patterns over time.
2. Product Analysis: Analyse the distribution of product categories, sizes, and quantities sold to identify popular products.
3. Fulfilment Analysis: Investigate the fulfilment methods used and their effectiveness in delivering orders.
4. Customer Segmentation: Segment customers based on their buying behaviour, location, and other relevant factors.
5. Geographical Analysis: Explore the geographical distribution of sales, focusing on states and cities.
6. Business Insights: Provide actionable insights and recommendations based on the analysis to optimize sales strategies, improve customer satisfaction, and enhance overall business performance.

Deliverables

1. Comprehensive Analysis Report: Summarizes key findings, insights, and recommendations.
2. Visualizations: Charts and graphs illustrating various aspects of the data analysis.
3. Insights: Insights on product preferences, customer behaviour, and geographical sales distribution.
4. Recommendations: Suggestions for improving sales strategies, inventory management, and customer service.

Step-by-Step Guide

1. Data Loading and Initial Inspection

- Objective: Load the dataset and inspect its structure.
- Steps:
 1. Import necessary libraries.
 2. Read the CSV file using `pandas`.
 3. Print the first few rows, column names, data types, and check for missing values.

2. Data Cleaning

- Objective: Clean the data to ensure it is ready for analysis.
- Steps:
 1. Handle missing values by dropping rows with null values.
 2. Convert the 'date' column to a datetime format.
 3. Inspect the cleaned data to confirm changes.

3. Sales Trends Over Time

- Objective: Analyse sales trends over time.
- Steps:
 1. Extract month and year from the 'date' column.
 2. Group data by month and calculate total sales.
 3. Plot the monthly sales trends.

4. Product Analysis

- Objective: Identify popular products based on category, size, and quantity sold.
- Steps:
 1. Calculate the distribution of product categories.
 2. Plot the distribution of product categories.
 3. Calculate the distribution of product sizes.
 4. Plot the distribution of product sizes.

5. Fulfilment Analysis

- Objective: Investigate the effectiveness of different fulfilment methods.

- Steps:

1. Calculate the distribution of fulfilment methods.
2. Plot the distribution of fulfilment methods.

6. Customer Segmentation

- Objective: Segment customers based on their buying behaviour and location.

- Steps:

1. Calculate the distribution of customer locations based on the 'ship-city' column.
2. Plot the distribution of customer locations.

7. Geographical Analysis

- Objective: Explore the geographical distribution of sales.

- Steps:

1. Group data by 'ship-state' and calculate total sales.
2. Plot the state-wise sales distribution.

8. Insights and Recommendations

- Objective: Provide actionable insights and recommendations based on the analysis.

- Steps:

1. Summarize key findings from each analysis section.
2. Formulate recommendations to optimize sales strategies, inventory management, and customer service.

Code Implementation

```
```python

import pandas as pd
import matplotlib.pyplot as plt

Define data types for columns
dtype_dict = {
 'index': 'float64',
 'Order ID': 'object',
 'date': 'object',
 'Status': 'object',
 'Fulfilment': 'object',
 'Sales Channel': 'object',
 'ship-service-level': 'object',
 'category': 'object',
 'Size': 'object',
 'Courier Status': 'object',
 'Qty': 'float64',
 'INR': 'object',
 'amount': 'float64',
 'ship-city': 'object',
 'ship-state': 'object',
 'ship-postal-code': 'float64',
 'ship-country': 'object',
 'B2B': 'object',
 'fulfilled-by': 'object',
 'New': 'float64',
 'PendingS': 'float64'
}
```

```
Load the dataset

file_path = 'report.csv'

data = pd.read_csv(file_path, dtype=dtype_dict, low_memory=False)

Display column names and data types

print("Column names:", data.columns)

print("Data types:", data.dtypes)

Handle missing values

data = data.dropna()

Convert 'date' column to datetime

data['date'] = pd.to_datetime(data['date'], errors='coerce')

Display cleaned data information

print("Cleaned data info:")

print(data.info())

Check for any rows with null dates after conversion

data = data.dropna(subset=['date'])

Sales trends over time

data['month'] = data['date'].dt.to_period('M')

monthly_sales = data.groupby('month')['amount'].sum()
```

```
if not monthly_sales.empty:
```

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

```
 monthly_sales.plot(kind='line')
```

```
 plt.title('Monthly Sales Trends')
```

```
 plt.xlabel('Month')
```

```
 plt.ylabel('Sales Amount')
```

```
 plt.show()
```

```
Product Analysis
```

```
product_categories = data['category'].value_counts()
```

```
if not product_categories.empty:
```

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

```
 product_categories.plot(kind='bar')
```

```
 plt.title('Product Category Distribution')
```

```
 plt.xlabel('Product Category')
```

```
 plt.ylabel('Count')
```

```
 plt.show()
```

```
Fulfillment Analysis
```

```
fulfillment_methods = data['Fulfilment'].value_counts()
```

```
if not fulfillment_methods.empty:
```

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

```
 fulfillment_methods.plot(kind='bar')
```

```
 plt.title('Fulfillment Method Distribution')
```

```
 plt.xlabel('Fulfillment Method')
```

```
 plt.ylabel('Count')
```

```
 plt.show()
```

```
Customer Segmentation
```

```
customer_location = data['ship-city'].value_counts()
```

```
if not customer_location.empty:
```

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

```
 customer_location.plot(kind='bar')
```

```
 plt.title('Customer Location Distribution')
```

```
 plt.xlabel('Location')
```

```
 plt.ylabel('Count')
```

```
 plt.show()
```

```
Geographical Analysis
```

```
state_sales = data.groupby('ship-state')['amount'].sum()
```

```
if not state_sales.empty:
```

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

```
 state_sales.plot(kind='bar')
```

```
 plt.title('State-wise Sales Distribution')
```

```
 plt.xlabel('State')
```

```
 plt.ylabel('Sales Amount')
```

```
 plt.show()
```

```
...
```



## **Summary of Findings**

1. Sales Trends: Monthly sales trends were analysed to identify peak sales periods.
2. Product Preferences: Distribution of product categories and sizes was examined to determine popular products.
3. Fulfilment Methods: Analysis of different fulfilment methods and their effectiveness.
4. Customer Locations: Geographical distribution of customers was mapped to identify key markets.
5. State Sales: Sales distribution across states was analysed to understand regional performance.

## **Recommendations**

1. Sales Strategy: Focus marketing efforts during peak sales periods identified in the trends analysis.
2. Inventory Management: Prioritize stocking popular product categories and sizes.
3. Fulfilment Optimization: Improve fulfilment methods that show lower effectiveness.
4. Customer Targeting: Tailor marketing campaigns to key customer locations identified in the analysis.
5. Regional Focus: Allocate resources to states with higher sales to maximize revenue potential.

This detailed analysis provides a comprehensive overview of the Amazon sales data, helping to inform business decisions and optimize operations for improved performance and customer satisfaction.