**Exploratory Analysis of Amazon Product Data and Customer Insights**

**Project Overview**

This project explores key factors that influence customer behavior, product performance, and satisfaction on the Amazon marketplace. By performing **Exploratory Data Analysis (EDA)** on an Amazon product dataset, the project aims to identify:

* Patterns in product ratings and customer feedback
* The impact of factors such as price, category, review count, and rating
* Trends and signals that influence high-performing or low-performing products
* Insights that can inform sellers about customer preferences and expectations

**🎯 Problem Definition**

To enhance product visibility, customer satisfaction, and market strategy, businesses must understand what drives product success and user engagement. This project attempts to answer:

* What are the key variables influencing product ratings and reviews?
* Which product categories tend to receive higher or lower satisfaction?
* What trends are visible across price points, review counts, and star ratings?

This project aims to uncover the driving factors behind product success on Amazon by analyzing structured product-level data, including variables such as product title, rating, price, category, review count, and textual review snippets. The goal is to extract meaningful patterns that explain what makes a product successful in terms of customer satisfaction and visibility.

**📁 Dataset Selection**

* **File Name**: Amazon\_Products.csv *(replace with actual filename if different)*
* **Data Type**: Structured tabular data
* **Data Size**: ~50,000 rows × ~10 columns *(adjust as per your data)*
* **Source**: [Kaggle / Amazon Product Reviews Dataset](https://www.kaggle.com/) *(or specify actual link)*
* **Content**: Includes product title, price, rating, number of reviews, category, and review text snippets

**Data Summary**

This dataset contains thousands of Amazon product listings across multiple categories. It includes structured information about each product’s name, price, star rating, number of customer reviews, and brief review content. The data is useful for performing e-commerce analytics, customer sentiment analysis, and market trend evaluation.

| **Column Name** | **Description** |
| --- | --- |
| Product\_ID | Unique identifier for each product |
| Product\_Title | Title or name of the product |
| Category | Category under which the product is listed |
| Price | Product price in USD |
| Rating | Average customer rating (1 to 5 scale) |
| Review\_Count | Number of customer reviews |
| Review\_Text | Snippet of customer feedback or review |
| Availability | Product availability (In Stock / Out of Stock) |
| Brand | Brand name of the product |
| Discount | Discount offered (if available) |

**🧪 Step 1: Data Loading and Initial Overview**

**1.1 📥 Importing Required Libraries**

python

CopyEdit

import pandas as pd

import numpy as np

**1.2 📂 Loading the Dataset**

python

CopyEdit

df = pd.read\_csv("Amazon\_Products.csv")

df.head()

**1.3 🔎 Dataset Overview**

**1.3.1 Dataset Shape**

python

CopyEdit

print ("Dataset contains:", df.shape[0], "rows and", df.shape[1], "columns")

**1.3.2 Column Data Types**

python

CopyEdit

df.dtypes

Likely:

* Product\_Title, Category, Brand, Availability → categorical
* Price, Rating, Review\_Count, Discount → numeric
* Review\_Text → text (object)

**1.3.3 Previewing the First 5 Rows**

python

CopyEdit

df.head()

**1.3.4 Previewing the Last 5 Rows**

python

CopyEdit

df.tail()

**1.3.5 DataFrame Structure Overview**

python

CopyEdit

df.info()

**1.3.6 📊 Summary Statistics**

python

CopyEdit

df.describe()

**1.3.7 Checking for Missing Values**

python

CopyEdit

df.isnull().sum()

**1.3.8 Unique Value Counts per Column**

python

CopyEdit

df.nunique()

**1.3.9 Checking for Duplicate Values**

python

CopyEdit

df.duplicated().sum()

**1.3.10 Preview Unique Values in Object Columns**

python

CopyEdit

object\_cols = df.select\_dtypes(include='object').columns

for col in object\_cols:

print(f"\n🟨 Unique values in '{col}':")

print(df[col].unique())