

Amazon Electronics Sales & Product Insights Project Report

Project Overview

This project focuses on analyzing the Amazon Electronics Products Sales Dataset to gain meaningful insights into product performance, pricing strategies, and customer satisfaction. The cleaned and analyzed data is visualized using Python and Tableau, targeting key metrics relevant to product management.

Objectives

- Understand sales performance across product categories.
- Analyze pricing strategy and discount impact.
- Evaluate customer satisfaction through ratings and reviews.
- Identify top-performing and underperforming products.

Data Cleaning & Preparation

- **Handled Missing Values:** Removed or imputed missing data.
- **Removed Duplicates:** Ensured data integrity by eliminating duplicate product entries.
- **Data Transformation:** Converted data types and created calculated fields for analysis.
- **Filtered Data:** Focused on relevant fields like product name, category, price, discount, rating, and review count.

Key Metrics & KPIs

1. **Sales Performance Analysis:**
 - Top 10 Most Reviewed Products
 - Total Reviews by Category
2. **Pricing Strategy Analysis:**
 - Average Price by Category
 - Top 5 Products with Highest Discounts
 - Discount Percentage Impact on Reviews
3. **Customer Satisfaction Metrics:**
 - Average Rating by Category
 - Top 5 Highest Rated Products
 - High Review, Low Rating Products
4. **Category-Level Insights:**
 - Value-for-Money Categories
 - Combined View of Price, Rating, and Discount

Visualizations in Python

- **Bar Plots:** Category-level analysis of price, rating, and discounts.
- **Scatter Plots:** Relationship between discount percentage and reviews.
- **Pie Charts:** Proportion of value-for-money categories.

Tableau Dashboard Layout

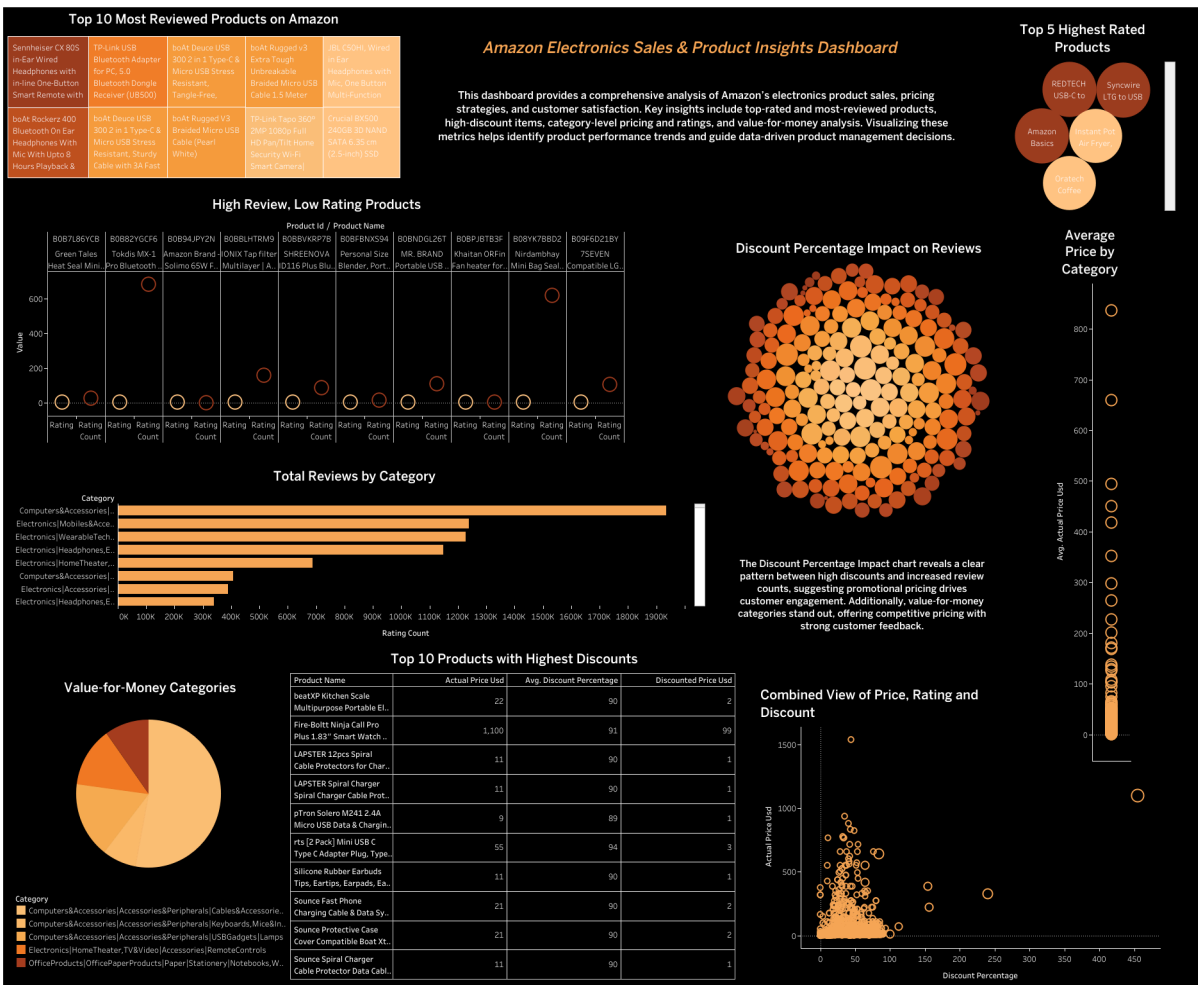
- **Sheets:** Separate sheets for individual KPIs and metrics.
- **Combined Dashboard:** Interactive view with filters and highlights.
- **Styling:** Consistent color themes and clear labeling.
- **Interactivity:** Drill-down capabilities and hover tooltips.

Insights & Recommendations

- **Top Products:** High-rated and well-reviewed products can be promoted more.
- **Pricing Strategy:** Products with high discounts but low ratings need quality checks.
- **Customer Preferences:** Categories with high ratings and reviews indicate popular segments.
- **Value-for-Money:** Balancing price, discount, and satisfaction ensures better product positioning.

Conclusion

This project provides a data-driven approach to understanding product performance on Amazon's electronics category. The insights generated are valuable for making informed product management and pricing decisions.



Appendix:

CODE:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
#load the dataset
df = pd.read_csv("amazon.csv")
df.head
#initial data inspection
print("Initial Data Info:\n")
print(df.info())
print("\n Initial Data Description:\n")
print(df.describe(include="all"))
print("\nMissing values Before Cleaning:\n")
print(df.isnull().sum())
#cleaning column names
df.columns = df.columns.str.strip().str.lower().str.replace(' ','_')
df.columns
# Converting price and discount columns to numeric
df['actual_price'] = df['actual_price'].str.replace('₹', '').str.replace(',', '').astype(float)
df['discounted_price'] = df['discounted_price'].str.replace('₹', '').str.replace(',', '').astype(float)
df['discount_percentage'] = df['discount_percentage'].str.replace('%', '').astype(float)
#changing ruppes to dollars
conversion_rate = 0.011 # 1 INR = 0.011 USD (example rate)

# Create a new column for prices in dollars
df['actual_price_usd'] = (df['actual_price'] * conversion_rate).round(2)
df['discounted_price_usd'] = (df['discounted_price'] * conversion_rate).round(2)
Df['actual_price_usd']
Df['discounted_price_usd']
Df['discount_percentage']
#dropping inr columns
columns_to_drop = ['actual_price', 'discounted_price']
df.drop(columns=columns_to_drop, axis=1, inplace=True)
df.head()
# Converting rating to numeric
df['rating'] = pd.to_numeric(df['rating'], errors='coerce')
# Handling missing values
df['rating'].fillna(df['rating'].mean(), inplace=True)
df['rating_count'].fillna(0, inplace=True)
# Removing duplicates
df.drop_duplicates(subset='product_id',inplace=True)
# Summary after cleaning
print("\nData Info After Cleaning:\n")
print(df.info())
print("\nData Description After Cleaning:\n")
print(df.describe(include='all'))
print("\nMissing Values After Cleaning:\n")
```

```

print(df.isnull().sum())
df.to_csv('cleaned_amazon.csv', index=True)
# Sales Performance Visualizations
df['rating_count'] = pd.to_numeric(df['rating_count'], errors='coerce')
df['rating_count'].fillna(0, inplace=True)

plt.figure(figsize=(10, 6))
sns.barplot(x='rating_count', y='product_name', data=df.sort_values('rating_count',
ascending=False).head(5))
plt.title("Top 10 Best-Selling Products")
plt.show()

df['revenue'] = df['discounted_price_usd'] * df['rating_count']
plt.figure(figsize=(10, 6))
sns.barplot(x='revenue', y='product_name', data=df.sort_values('revenue',
ascending=False).head(5))
plt.title("Top 10 Revenue-Contributing Products")
plt.show()
# Pricing Strategy Visualizations
plt.figure(figsize=(8, 6))
sns.scatterplot(x='actual_price_usd', y='rating', data=df)
plt.title("Price vs Rating")
plt.show()

plt.figure(figsize=(8, 6))
sns.scatterplot(x='discount_percentage', y='rating_count', data=df)
plt.title("Discount Percentage vs Rating Count")
plt.show()
# Customer Satisfaction Visualizations
plt.figure(figsize=(8, 6))
sns.histplot(df[df['rating'] >= 4.5]['rating'], bins=10, kde=True)
plt.title("Distribution of Top Rated Products")
plt.show()

plt.figure(figsize=(10, 6))
sns.barplot(x='rating', y='product_name', data=df[(df['rating'] < 3.0) & (df['rating_count'] >
100)].sort_values('rating_count', ascending=False))
plt.title("Products with High Reviews but Low Ratings")
plt.show()
# Category-Level Visualizations
top_5_categories = df['category'].value_counts().head(5)
sns.countplot(y='category', data=df[df['category'].isin(top_5_categories.index)])
plt.xticks(rotation=45)
plt.title('Top 5 Product Categories')
plt.show()
# Grouping and summarizing data
category_summary = df.groupby('category').agg({
    'rating': 'mean',

```

```
'actual_price_usd': 'mean',  
'discount_percentage': 'mean'  
}).reset_index()
```

```
# Top 5 categories by average rating
```

```
top_5_rating = category_summary.sort_values('rating', ascending=False).head(5)
```

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

```
sns.barplot(x='rating', y='category', data=top_5_rating, palette='viridis')
```

```
plt.title("Top 5 Categories by Average Rating")
```

```
plt.xlabel('Average Rating')
```

```
plt.ylabel('Category')
```

```
plt.show()
```

```
# Top 5 categories by average price
```

```
top_5_price = category_summary.sort_values('actual_price_usd', ascending=False).head(5)
```

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

```
sns.barplot(x='actual_price_usd', y='category', data=top_5_price, palette='magma')
```

```
plt.title("Top 5 Categories by Average Price")
```

```
plt.xlabel('Average Price (USD)')
```

```
plt.ylabel('Category')
```

```
plt.show()
```

```
# Top 5 categories by average discount percentage
```

```
top_5_discount = category_summary.sort_values('discount_percentage',  
ascending=False).head(5)
```

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

```
sns.barplot(x='discount_percentage', y='category', data=top_5_discount, palette='cividis')
```

```
plt.title("Top 5 Categories by Average Discount Percentage")
```

```
plt.xlabel('Average Discount Percentage')
```

```
plt.ylabel('Category')
```

```
plt.show()
```

VISUALIZATIONS:

