**Flipkart Products Data Analysis**

**1. Introduction:**

The Flipkart products collection offers a comprehensive array of information about various products available on the Flipkart e-commerce platform. This collection includes attributes such as product names, categories, prices, ratings, and reviews, providing a valuable resource for analysing consumer behaviour, market trends, and product performance. As one of India's leading online marketplaces, Flipkart provides insights into the e-commerce landscape, allowing businesses and researchers to make informed decisions based on data-driven insights. This analysis aims to explore the data to uncover patterns and trends that can help understand the factors influencing product sales and customer preferences.

**2. Objectives:**

**1. Data Exploration:** Conduct an exploratory data analysis (EDA) to understand the structure, summary statistics, and key features.

**2. Price Analysis:** Analyse the pricing strategies of different product categories and identify any trends in product pricing.

**3. Customer Ratings**: Evaluate the impact of customer ratings on product sales and overall customer satisfaction.

**4. Category Performance:** Assess the performance of various product categories based on sales volume, ratings, and customer feedback.

**5. Market Trends:** Identify emerging trends in consumer preferences and market demand based on the analysis of the product information.

**3. Methodology**

The project will follow a structured approach:

• **Data Collection:** The dataset will be sourced from Cagle, ensuring it is well documented and reliable.

• **Exploratory Data Analysis (EDA):** Handle missing data using imputation techniques, Summarise the dataset using descriptive statistics and Create visualizations (e.g., histograms, box plots, correlation heat maps) to understand feature distributions and interactions.

• **Visualization:** Generate visual representations of findings to support conclusions.

• **Reporting:** Compile analysis, results, and insights into a detailed report.

**4. Tools and Technologies**

The project will utilize the following tools and technologies:

* **Programming Language:** Python
* **Libraries:** Pandas, NumPy, Matplotlib, Seaborn, Scikit-learn.
* **IDE:** Jupyter Notebook or any Python-compatible Integrated Development Environment (IDE)
* **Data Source:** Kaggle Repository (Flipkart Products Dataset)

**5. Exploratory Data analysis (EDA):**

**Importing Necessary Libraries**

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

import numpy as np

**Loading and Inspecting the Dataset**

1. Load the dataset using `pd.read\_csv`.

2. Display the first few rows of the data using `df.head()`.

3. View dataset information with `df.info()` to understand data types and null values.

df = pd.read\_csv('flipkart.csv')

df.head()

df.info()

**Identifying and Handling Duplicates**

1. Check for duplicate rows using `df.duplicated()`.

2. Count total duplicate rows with `df.duplicated().sum()`.

df.duplicated()

df.duplicated().sum()

**Handling Missing Values**

1. For the `brand` column: Fill missing values with the most frequent value (mode).

2. For `retail\_price` and `discounted\_price’: Fill missing values with their respective mean values.

mode\_value = df['brand'].mode()[0]

df['brand'].fillna(mode\_value, inplace=True)

mean\_value = df['retail\_price'].mean()

df['retail\_price'].fillna(mean\_value, inplace=True)

mean\_value = df['discounted\_price'].mean()

df['discounted\_price'].fillna(mean\_value, inplace=True)

3. Dropping Columns: Delete columns that still contain null values.

4. Updating Dataset Information: Check dataset information to confirm changes.

df.dropna(axis=1, inplace=True)

df.info()

**Converting to DateTime Format**

1. Convert `crawl\_timestamp` column to DateTime format.

2. Extract `year` and `month` information for analysis.

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

df['year'] = df['crawl\_timestamp'].dt.year

df['month'] = df['crawl\_timestamp'].dt.month

df.info()

**Basic Statistics**

View descriptive statistics to understand data distribution.

df.describe()

**6. Data Visualization**

**Scatter Plot: Product Rating vs Discounted Price**

This scatter plot shows the relationship between product ratings and discounted prices.

sns.scatterplot(x='product\_rating', y='discounted\_price', data=df)

plt.title('Product Rating vs Discounted Price')

plt.show()

**Price Analysis: Average Retail and Discounted Prices Across Categories**

1. Extract the main category from `product\_category\_tree`.

2. Calculate the discount percentage for each product.

3. Aggregate average retail price, discounted price, and discount percentage by category.

df['main\_category'] = df['product\_category\_tree'].str.split('>>').str[0].str.strip()

df['discount\_percentage'] = ((df['retail\_price'] - df['discounted\_price']) / df['retail\_price']) \* 100

price\_analysis = df.groupby('main\_category').agg(

avg\_retail\_price=('retail\_price', 'mean'),

avg\_discounted\_price=('discounted\_price', 'mean'),

avg\_discount\_percentage=('discount\_percentage', 'mean')

).reset\_index()

print(price\_analysis)

**Plot the top 10 categories by average discounted price to visualize price and discount trends**.

top\_categories = price\_analysis.nlargest(10, 'avg\_discounted\_price')

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

sns.scatterplot(data=top\_categories, x='avg\_discounted\_price', y='avg\_discount\_percentage', hue='main\_category', s=100)

plt.title('Top 10 Categories: Avg Discounted Price vs. Avg Discount Percentage')

plt.xlabel('Average Discounted Price (₹)')

plt.ylabel('Average Discount Percentage (%)')

plt.legend(title='Category', bbox\_to\_anchor=(1.05, 1), loc='upper left')

plt.tight\_layout()

plt.show()

**Market Trends: Average Price and Rating Over Time**

1. Create a `year\_month` column for monthly analysis.

2. Group by `year\_month` and calculate average discounted price and product rating.

3. Plot these trends over time.

df['year\_month'] = df['crawl\_timestamp'].dt.to\_period('M')

trend\_data = df.groupby('year\_month').agg({

'discounted\_price': 'mean',

'product\_rating': 'mean'

}).reset\_index()

plt.figure(figsize=(14, 6))

# Plot average discounted price

plt.subplot(1, 2, 1)

plt.plot(trend\_data['year\_month'].astype(str), trend\_data['discounted\_price'], marker='o', color='blue')

plt.title('Average Discounted Price Over Time')

plt.xlabel('Date')

plt.ylabel('Average Discounted Price')

plt.xticks(rotation=45)

plt.grid()

# Plot average product rating

plt.subplot(1, 2, 2)

plt.plot(trend\_data['year\_month'].astype(str), trend\_data['product\_rating'], marker='o', color='orange')

plt.title('Average Product Rating Over Time')

plt.xlabel('Date')

plt.ylabel('Average Product Rating')

plt.xticks(rotation=45)

plt.grid()

plt.tight\_layout()

plt.show()

**Category Performance: Sales vs Ratings**

1. Calculate average discounted price and product rating for each category.

2. Filter the top 10 categories by average discounted price.

3. Create a dual-axis bar plot to show sales performance versus product ratings.

category\_performance = df.groupby('main\_category').agg(

avg\_discounted\_price=('discounted\_price', 'mean'),

avg\_product\_rating=('product\_rating', 'mean')

).reset\_index()

top\_categories = category\_performance.nlargest(10, 'avg\_discounted\_price')

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

# Bar plot for average discounted price

ax1 = sns.barplot(x='main\_category', y='avg\_discounted\_price', data=top\_categories, color='lightblue')

ax1.set\_ylabel('Average Discounted Price (₹)')

ax1.set\_xlabel('Product Category')

ax1.set\_title('Category Performance: Sales vs Ratings')

# Rotate x-tick labels

plt.xticks(rotation=45, ha='right')

# Line plot for product ratings

ax2 = ax1.twinx()

sns.lineplot(x='main\_category', y='avg\_product\_rating', data=top\_categories, color='red', marker='o', ax=ax2)

ax2.set\_ylabel('Average Product Rating')

plt.tight\_layout()

plt.show()

**Correlation Analysis: Numerical Features**

1. Select numerical columns for correlation analysis.

2. Compute and visualize the correlation matrix to examine relationships between numerical features.

numerical\_data = df[['retail\_price', 'discounted\_price', 'product\_rating', 'year', 'month']]

correlation\_matrix = numerical\_data.corr()

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

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

plt.title('Correlation Matrix of Numerical Features')

plt.show()

**6. Conclusion:**

The analysis of Flipkart's product data reveals key trends in pricing, discounts, and product ratings across various categories. We observed that while discounts are prevalent, higher discounts don’t necessarily correlate with higher product ratings. Category-level analysis highlighted certain product categories with more aggressive discounting strategies, likely tied to promotional events or inventory clearance. Time-based trends in discounted prices and ratings suggest seasonal sales cycles, with noticeable spikes during particular months. Overall, these insights can help inform inventory and pricing strategies by pinpointing high-performing categories and optimal discount periods.