TOPIC:

FLIPKART PRODUCT DATASET ANALYSIS

PRESENTED BY:

- **SHIVI AGRAWAL (202410101150182)**
- **ALIZA AKBAR (202410101150222)**

GROUP NO.:

11

Flipkart Product Dataset Analysis

Project Overview

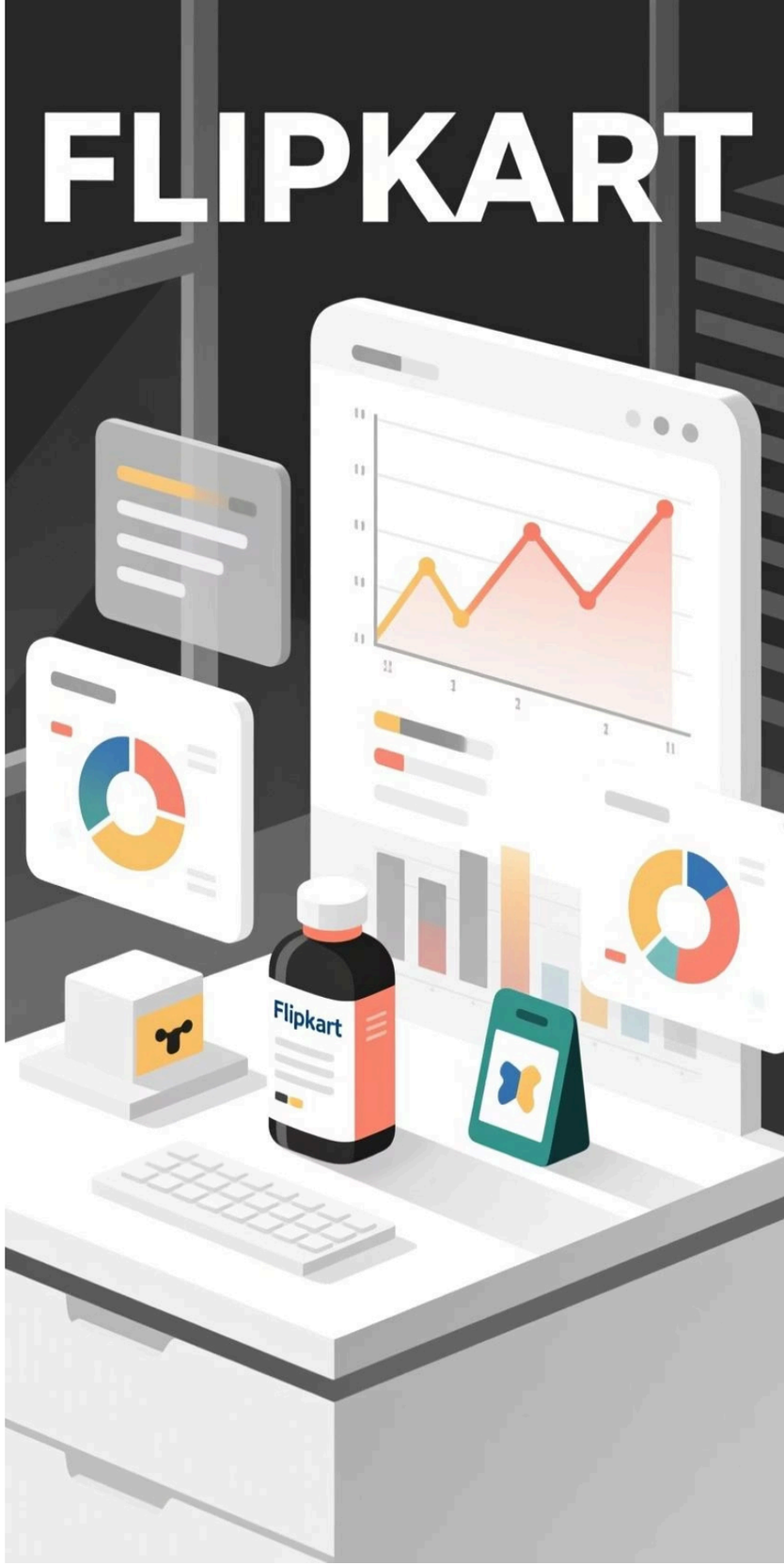
The Flipkart Product Dataset obtained from **Kaggle**, a popular open-source data platform, contains detailed information about various products, including attributes such as **MRP, selling price, discount, ratings, category, and seller details**. We want to look closely at this data to find out what it tells us. Our goal is to explore and understand this rich dataset using Exploratory Data Analysis (EDA) techniques.

Objective

Uncover hidden trends, understand pricing patterns, and evaluate data quality before any advanced analytics.

Methodology

Applying EDA techniques using Python libraries such as **Pandas, NumPy, and Matplotlib** to achieve data readiness and insight extraction.



Objective

A comprehensive EDA process ensures data integrity and prepares the ground for effective modeling and business decision-making.

Data Examination

Analyze the structure, identify missing/incorrect values, and clean the dataset for optimal accuracy.

Statistical Interpretation

Calculate and interpret fundamental statistical measures (mean, median, mode) to understand central tendencies and spread.

Group-Wise Analysis

Summarize and compare product data across key dimensions like categories and sellers using GroupBy methods.

Visual Insight Extraction

Employ visualizations (histograms, bar charts, scatter plots) to extract insights on pricing, discounts, and product ratings.

Import: Libraries and Load Data

The analysis relies on key Python libraries tailored for data science tasks. The foundation begins with correctly loading the raw data.

Pandas

Essential for data loading, manipulation, and high-level analytical tasks on structured data.

NumPy

Provides support for large, multi-dimensional arrays and matrices, crucial for numerical computations and handling null values.

Visual Libraries

Matplotlib is used for creating rich, informative statistical visualizations to uncover trends.

Data Ingestion

The raw dataset file is loaded into a Pandas **DataFrame** using `pd.read_csv()`. Initial inspection confirms that all columns are read correctly and the data structure is maintained.

```
#Import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

#Import csv file
Prod= pd.read_csv("/content/drive/MyDrive/Data analytics and reporting(DAR)/Flipkart Product Dataset.csv")
```

```
#View dimensions of data sets
print(Prod.shape)
```

```
(12041, 11)
```

Data Overview: Understanding the Dataset Structure

Initial exploration is vital to gauge the scale, types, and quality of the data we are working with.

1

Dataset Shape

Determines the total number of rows and columns, giving a precise measure of the dataset's volume.

2

Head()

Inspects the first rows to quickly verify data format, types, and column consistency.

3

Info()

Provides data types, non-null counts, helping to spot initial issues like incorrect types or pervasive missing data.

4

Describe()

Generates summary statistics (mean, quartiles, min/max) for all numerical features like price and discount.

5

Feature Columns

Explicitly lists features such as **MRP, selling_price, discount, product_rating,** and **categories**.

This foundational step confirms which fields are relevant and informs subsequent cleaning strategies.

Missing Values Analysis

Missing data can severely bias analysis. Identifying and treating these gaps is a critical stage in data preparation.

Identifying Gaps

The function `isnull().sum()` provides an essential column-wise count of all missing values, pinpointing columns that require attention.

```
#Check missing values
print(Prod.isnull().sum())
```

```
category_1           0
category_2           0
category_3           0
title               18
product_rating      80
selling_price       28
mrp                375
seller_name        216
seller_rating      214
description       7020
highlights        5481
dtype: int64
```

Strategic Imputation Methods

→ Deletion (`dropna()`)

Removes rows or entire columns that contain an excessive number of missing values (e.g., above 50% threshold).

→ Statistical Replacement (`fillna()`)

Replaces missing numeric values with the column's mean or median, and categorical values with the mode.

→ Propagation (`ffill` / `bfill`)

Uses Forward Fill (propagates the previous non-null observation) or Backward Fill (propagates the next non-null observation) for time-series or sequential data.

After treatment, a final check confirms the absence of null values, ensuring all downstream calculations are accurate.

Data Cleaning

Beyond missing values, effective data cleaning involves several adjustments to ensure consistency and usability of features.

Type Conversion

Ensuring columns are in the correct format, such as converting price fields from string-based representations to float data types.

Duplicate Handling

Identifying and removing duplicate rows, which often represent redundant or erroneous product listings, to maintain data integrity.

Standardization

Harmonizing categorical values by ensuring consistent casing and spelling (e.g., 'electronics' vs. 'Electronics').

Outlier Management

Reviewing and correcting extreme outliers that could disproportionately skew statistical results and analysis.

```
#Mean
Mean=Prod['product_rating'].mean()
print(Mean)
```

```
4.061616921662068
```

Basic Statistics: Central Tendency Measures

The core statistics provide a rapid summary of the dataset's numerical distribution, informing us about the typical values for key metrics.

Mean (Average)

The sum of all values divided by the count. Useful for understanding the average state, such as the **average selling price** across all products.

Median (Midpoint)

The middle value in an ordered set. Provides a robust measure of central tendency, as it is less sensitive to extreme outliers and skewed data.

Mode (Frequency)

The most frequently occurring value. Primarily used for categorical data to identify the **most common product category** or **seller**.

These measures together summarize the overall data pattern and highlight important characteristics regarding the spread and symmetry of values, essential for metrics like product rating and pricing.

Data Filtering and Sorting

Data manipulation techniques allow analysts to focus on subsets of high-interest data points, such as top-performing products or specific price ranges.

1

Filtering Data

Involves selecting specific rows based on defined conditions. Examples include isolating products where `product_rating >= 4` or focusing on items with a `selling_price > 500`.

2

Sorting Data

Arranging the dataset in ascending or descending order based on a specific feature, such as sorting by **selling_price** or **discount**.

By enabling focused analysis, these steps help identify high-value opportunities, compare pricing strategies within narrowly defined segments, and track top-rated inventory.

```
#Filtering data
filtered_product_rating= Prod['product_rating'] >= 4
filtered_product_rating
```

	product_rating
0	True
1	True
2	False
3	False
4	False
...	...
12036	False
12037	True
12038	True
12039	True
12040	False

12041 rows × 1 columns

dtype: bool

GroupBy Statistical Analysis

GroupBy is crucial for aggregating data across distinct segments, providing powerful comparative insights into category and seller performance.

GroupBy Mean

Calculates the average value (e.g., average selling price or rating) for each group (category/seller). **Identifies top-performing segments** on average.

GroupBy Median

Finds the middle value within each group. Provides a more **robust typical value** when outliers are present (e.g., median discount by category).

GroupBy Standard Deviation (Std)

Measures the variation or dispersion of values within a group. A higher Std implies greater inconsistency, such as **large price differences** among a seller's products.

This segmented analysis helps in understanding performance consistency, pricing stability, and overall variation across the e-commerce landscape.

Pivot Table Analysis

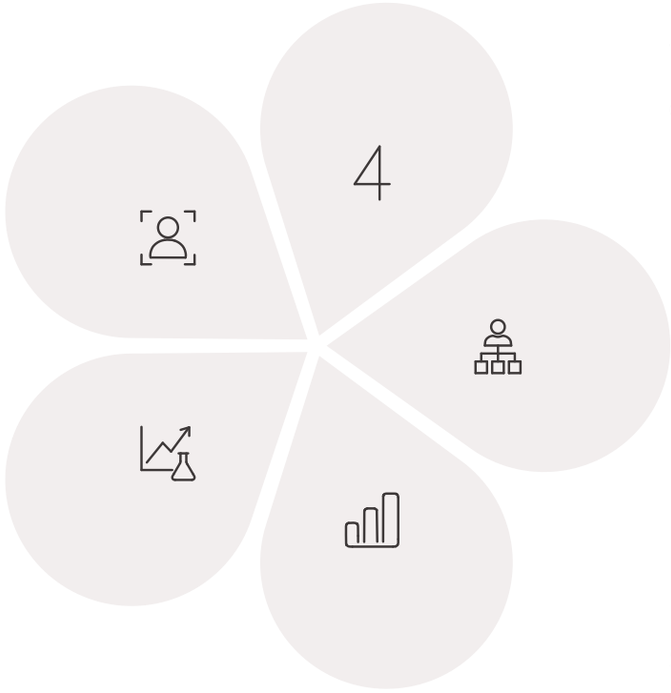
Pivot tables are essential tools for structuring data in a cross-tabular format, allowing analysts to quickly grasp relationships between multiple features.

Value Aggregation

Used to display aggregated metrics, such as the **average selling price**, within the intersecting cells.

Handling Nulls

Allows for replacement of empty cells with meaningful defaults (like 0 or a placeholder) for clearer visualization.



Row Index

Defines the main vertical dimension, typically a high-level grouping like **category_1**.

Column Grouping

Defines the horizontal dimension, useful for a secondary breakdown like **category_2**.

Multi-Axis View

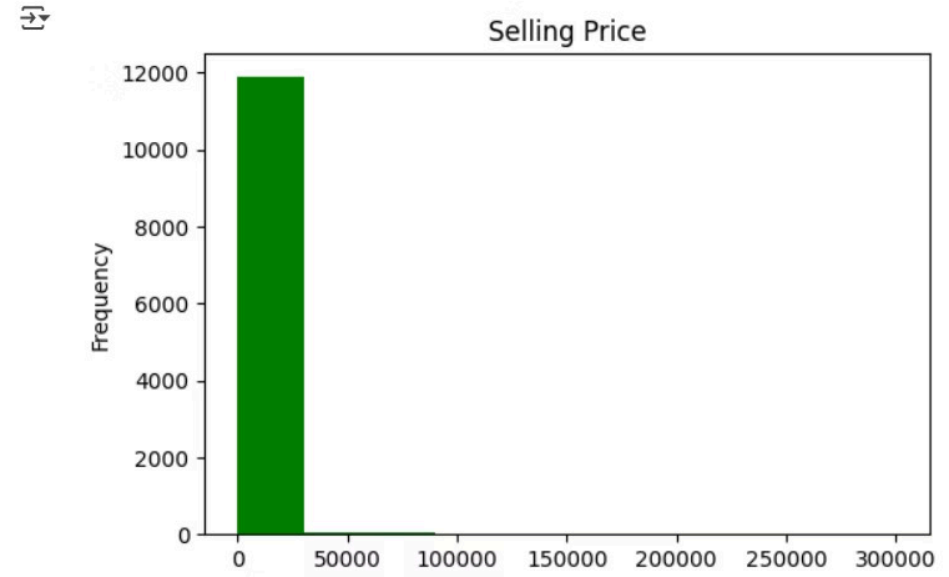
Enables a multi-dimensional comparison, showing how specific product combinations perform in terms of sales or ratings.

Pivot tables structure complex data to make comparative analysis across multiple variables straightforward and visually accessible.

Data Visualization

Effective data visualization transforms complex datasets into understandable graphics, revealing patterns, trends, and outliers at a glance.

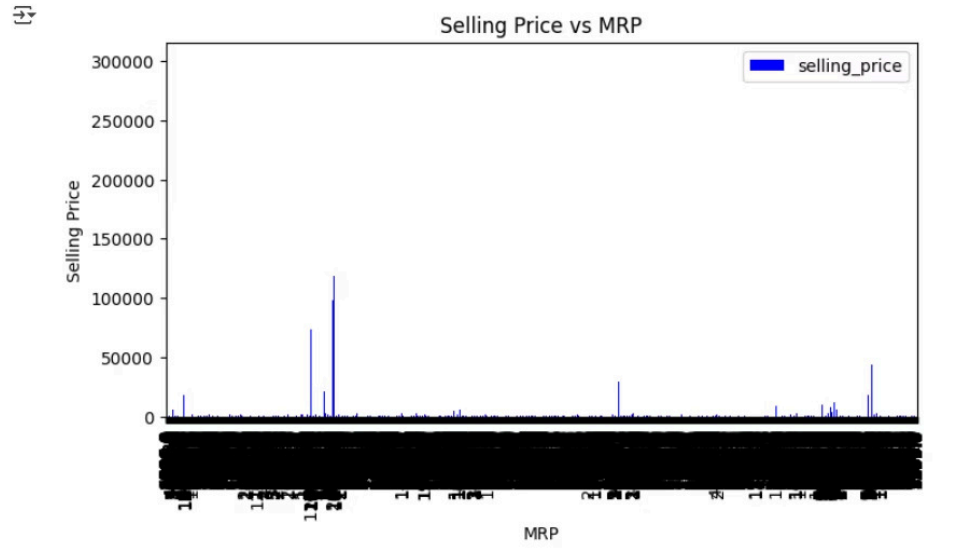
```
#Creates a histogram for the column selling_price
Prod['selling_price'].plot(kind="hist", figsize=(6,4), color="green")
# Adds a title
plt.title("Selling Price")
# Displays the plot
plt.show()
```



Histogram

Illustrates the distribution of numerical data, highlighting frequency, skewness, and potential outliers in metrics like **selling price** or **product ratings**.

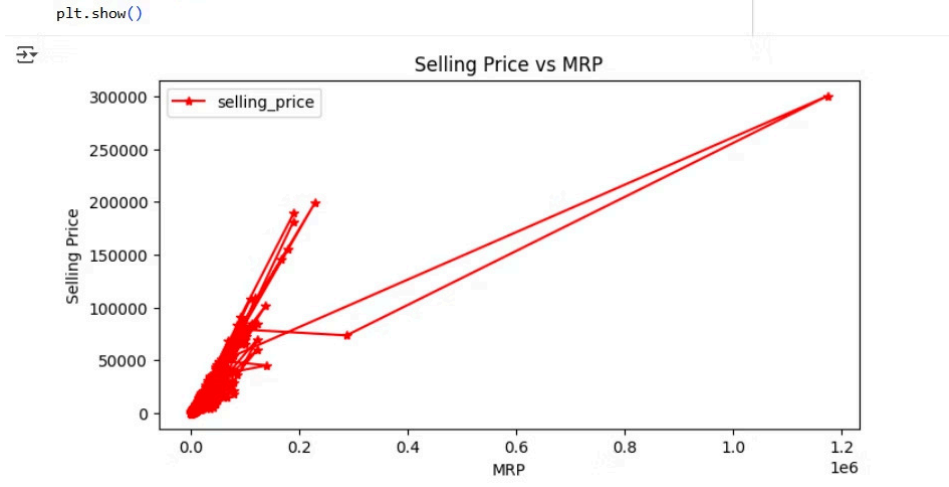
```
#Create a bar plot of selling_price vs mrp
Prod.plot(kind='bar', x='mrp', y='selling_price', figsize=(8,4), color='blue')
#Add labels and title
plt.xlabel("MRP")
plt.ylabel("Selling Price")
plt.title("Selling Price vs MRP")
#Display the plot
plt.show()
```



Bar Chart

Ideal for comparing categorical data, such as the **average selling price** or **product count** across different categories, to pinpoint top performers.

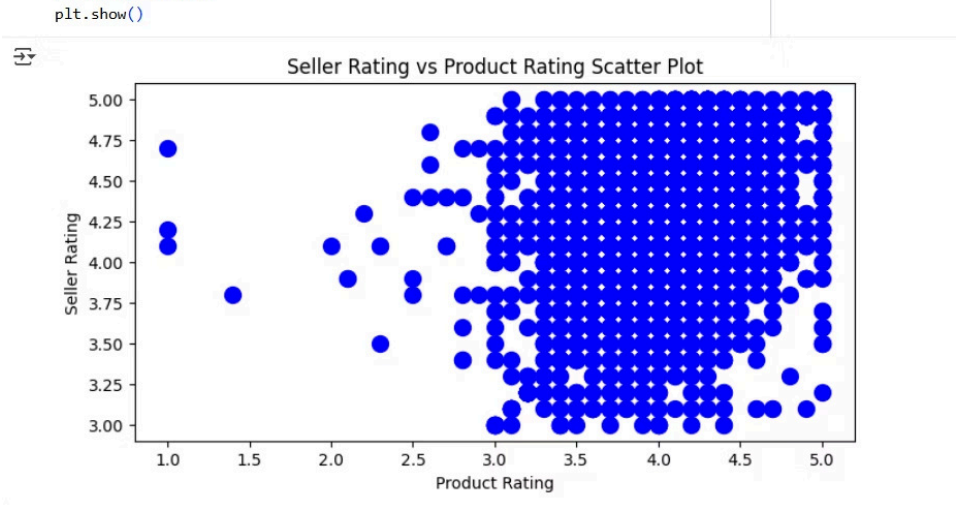
```
#Create a line plot of selling_price vs product_rating
Prod.plot(kind='line', x='mrp', y='selling_price', figsize=(8,4),marker="*", color='red', markersize=6)
# Add labels and title
plt.xlabel("MRP")
plt.ylabel("Selling Price")
plt.title("Selling Price vs MRP")
# Display the plot
plt.show()
```



Line Chart

Displays continuous data trends over time, allowing for easy observation of changes in average prices, ratings, or sales volume across periods.

```
#Create a scatter plot of seller_rating vs product_rating
Prod.plot(kind='scatter', x='product_rating', y='seller_rating', figsize=(8,4), color='blue', s=100)
#Add labels and title
plt.xlabel("Product Rating")
plt.ylabel("Seller Rating")
plt.title("Seller Rating vs Product Rating Scatter Plot")
# Display the plot
plt.show()
```



Scatter Plot

Reveals relationships and correlations between two numeric variables, for example, between **selling price** and **product rating**, to identify dependencies.

By employing clear titles, labeled axes, and consistent color schemes, these visualizations simplify interpretation and enhance the discovery of critical business insights.

Conclusion

Our Exploratory Data Analysis (EDA) of the Flipkart product dataset provided a comprehensive overview, revealing critical insights into market dynamics.

Analysis Summary

- Exploratory Data Analysis (EDA) leveraging Pandas and Matplotlib for data processing.
- Comprehensive missing data handling via drop and fill techniques.
- Basic statistics unveiled patterns in pricing, ratings, and discounts.
- GroupBy analysis delivered performance insights across categories and sellers.
- Trends clarified through pivot tables and data visualizations.

Key Insights Gained

- Significant variance in average selling prices and ratings across product categories.
- Wide disparity in discount levels between sellers and product categories.
- Initial data quality challenges necessitated rigorous cleaning and standardization.

This EDA successfully clarified the dataset’s underlying structure, quality, and prevalent market trends. It establishes a robust analytical foundation for advanced modeling and informed business strategy formulation.

Thank You!

Thank you for your attention and for listening to our presentation on the Flipkart Product Dataset Analysis. We hope the key insights from our data exploration give you a strong foundation for making smart decisions and for looking into things more in the future.