```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

Unveiling Flipkart's Product Landscape: A Comprehensive Analysis of Categories, Pricing, Brands, and Ratings

Project Summary

In our project, we began by exploring and preprocessing the dataset, ensuring its readiness for analysis. We then delved into uncovering insights from the data. We examined price trends over time, revealing how retail and discounted prices changed each month. Simultaneously, we investigated rating trends over time, tracking average product ratings through the months. A combination analysis showcased the highest-rated brand within a selected category. By identifying the most selling categories, we pinpointed the top 20 categories driving sales. Additionally, we examined seasonal trends and customer preferences, unveiling which product categories thrived during festive seasons. These insights were presented using intuitive visualizations, empowering us to make data-driven decisions.

Project Objective:

- Product Category Analysis: Identify the most popular product categories on Flipkart.
- Pricing and Discounts Analysis : Investigate the relationship between retail prices, discounted prices, and product categories. Analyze the average discounts offered for different Products.
- Analyzing Price and Rating Trends Over Time: Unveiling the Evolution of Product Costs and Customer Satisfaction.

Seasonal Trends and Customer Preferences:

- Are there specific months when customers prefer to purchase products with higher discounts?
- How do different product categories perform during seasonal sales or festive seasons?

```
df = pd.read_csv('flipkart_com-ecommerce_sample.csv')
```

Data Exploration and Cleaning

In this section, we conducted an exploratory analysis of the Flipkart dataset and performed necessary data cleaning steps to prepare it for further analysis. The goal was to gain insights into the dataset's structure, identify missing values, and address any inconsistencies.

Exploratory Data Analysis (EDA)

We began by conducting an exploratory data analysis to understand the basic characteristics of the dataset. Key steps in our EDA included:

- Loading the dataset into a pandas DataFrame.
- Displaying the first few rows of the dataset using the head() function to get an initial overview of the data.
- Using the info() function to obtain information about the columns, data types, and missing values.
- Calculating summary statistics of numerical columns using the describe() function.

Data Cleaning

Based on our EDA, we performed the following data cleaning actions:

- Converted the 'crawl_timestamp' column to a datetime format using pd.to_datetime() to facilitate time-based analysis.
- Examined the 'brand' column, which contained a significant number of missing values.
- Replaced missing 'brand' values with the value 'Missing' using the fillna() function to facilitate consistent analysis.

```
df.shape
```

```
(20000, 15)
```

```
df.columns
```

```
Index(['uniq_id', 'crawl_timestamp', 'product_url', 'product_name',
       'product_category_tree', 'pid', 'retail_price',
'discounted_price',
       'image', 'is_FK_Advantage_product', 'description',
'product_rating',
       'overall_rating', 'brand', 'product_specifications'],
      dtype='object')
```

```
fdf = df.copy()
```

```
fdf.columns
```

```
Index(['uniq_id', 'crawl_timestamp', 'product_url', 'product_name',
       'product_category_tree', 'pid', 'retail_price',
'discounted_price',
       'image', 'is_FK_Advantage_product', 'description',
'product_rating',
```

```
       'overall_rating', 'brand', 'product_specifications'],
      dtype='object')
```

```
# Convert 'crawl_timestamp' column to datetime
fdf['crawl_timestamp'] = pd.to_datetime(df['crawl_timestamp'],
format='%Y-%m-%d %H:%M:%S %z')

fdf['brand'].fillna('Missing', inplace=True)

fdf.isnull().sum()
```

```
uniq_id                      0
crawl_timestamp              0
product_url                  0
product_name                 0
product_category_tree        0
pid                          0
retail_price                78
discounted_price            78
image                        3
is_FK_Advantage_product      0
description                  2
product_rating               0
overall_rating               0
brand                        0
product_specifications      14
dtype: int64
```

```
print(fdf.dtypes)
```

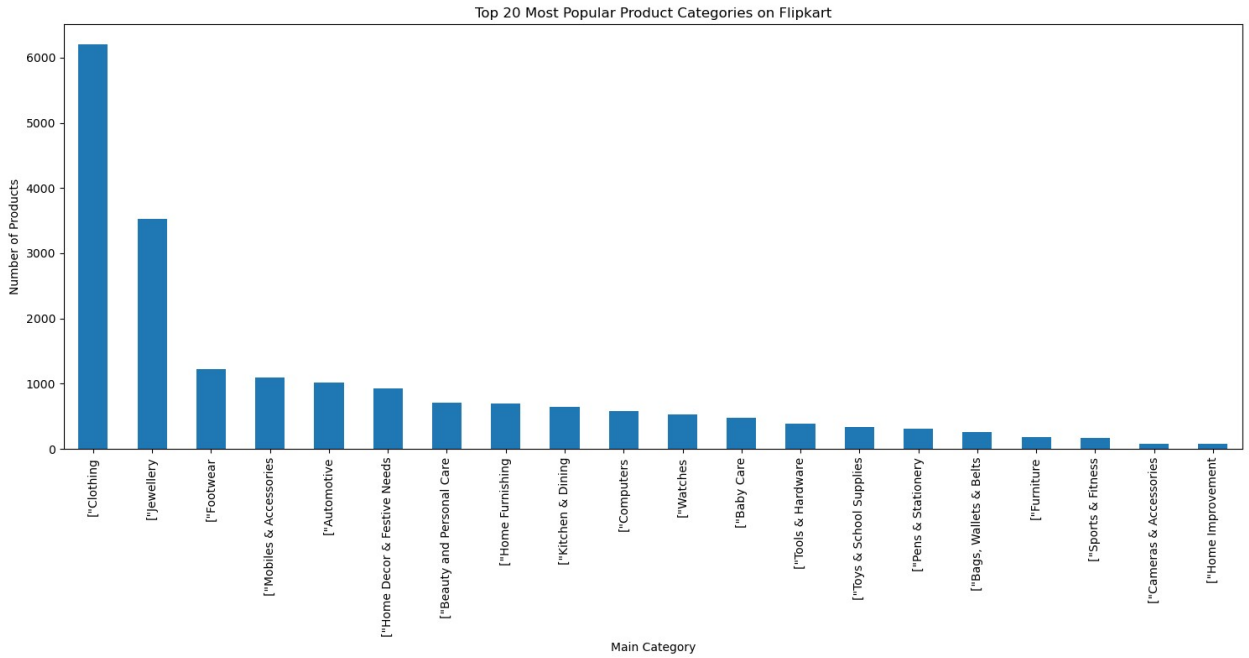
```
uniq_id                                  object
crawl_timestamp            datetime64[ns, UTC]
product_url                              object
product_name                             object
product_category_tree                    object
pid                                      object
retail_price                            float64
discounted_price                        float64
image                                    object
is_FK_Advantage_product                    bool
description                              object
product_rating                           object
overall_rating                           object
brand                                    object
product_specifications                   object
dtype: object
```

Problem Solving: Analytical Questions

In this section, we tackle a series of analytical questions based on the Flipkart dataset. Through data manipulation, calculations, and visualization, we aim to derive insights and answers to these questions.

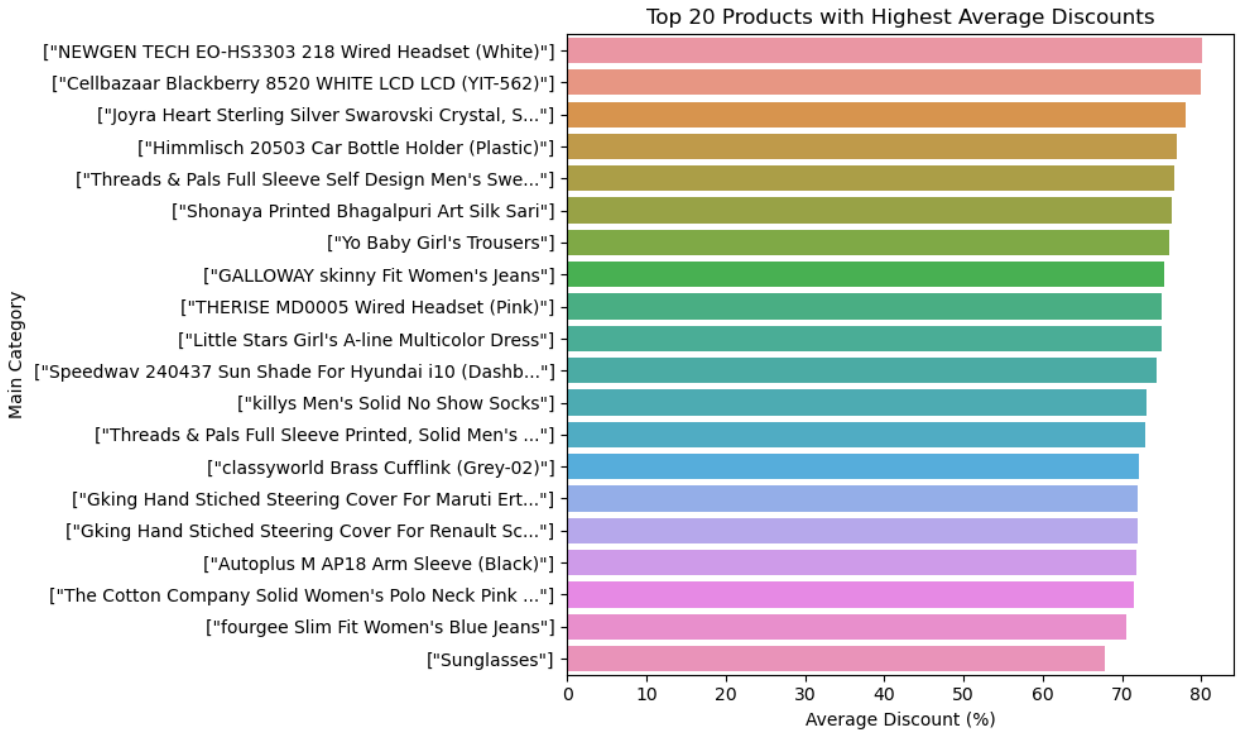
- Identify the most popular product categories on Flipkart. So, we will take top 20 categories.

```
# Extract main category from 'product_category_tree'
fdf['main_category'] = fdf['product_category_tree'].apply(lambda x:
x.split('>>')[0].strip())

# Count occurrences of each main category
category_counts = fdf['main_category'].value_counts()

# Select the top 20 most popular categories
top_categories = category_counts.head(20)

# Visualize the distribution of top 20 main categories
plt.figure(figsize=(15, 8))
top_categories.plot(kind='bar')
plt.title('Top 20 Most Popular Product Categories on Flipkart')
plt.xlabel('Main Category')
plt.ylabel('Number of Products')
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```



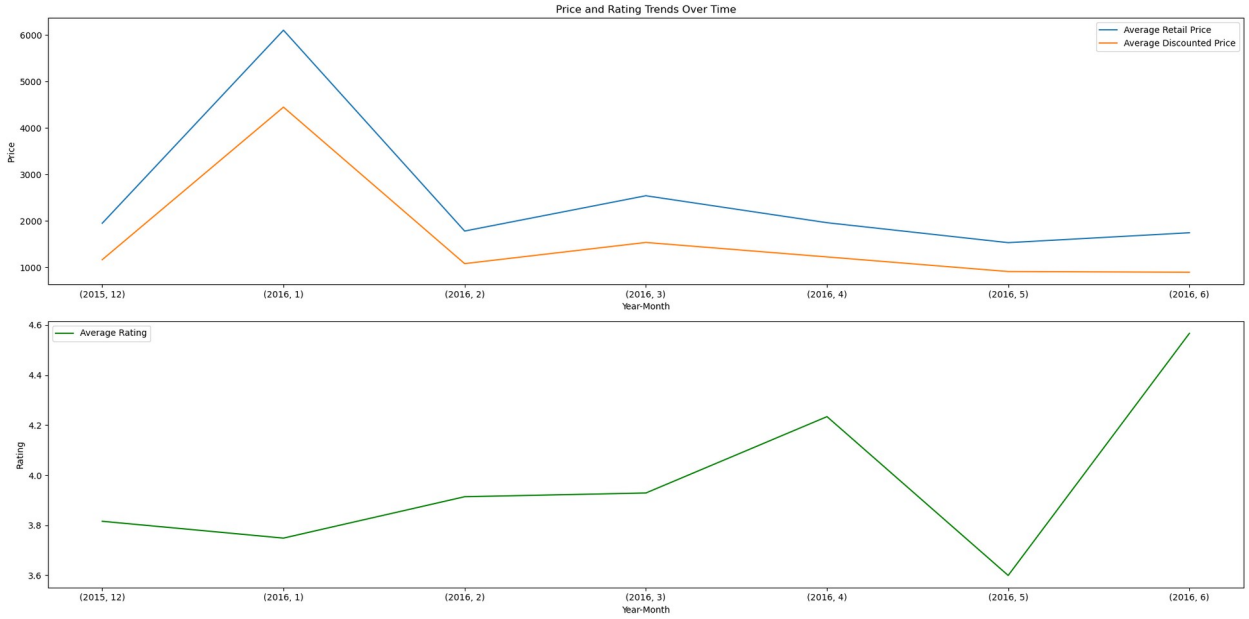
- Pricing and Discounts Analysis : Investigate the relationship between retail prices, discounted prices, and product categories. Analyze the average discounts offered for different Products.

```
# Calculate discount percentage for each product
fdf['discount_percentage'] = ((fdf['retail_price'] -
fdf['discounted_price']) / fdf['retail_price']) * 100

# Group by main category and calculate average discounts
category_avg_discounts = fdf.groupby('main_category')
['discount_percentage'].mean().reset_index()

# Select the top 10 categories with the highest average discounts
top_categories = category_avg_discounts.nlargest(20,
'discount_percentage')

# Create a horizontal bar plot using Seaborn
plt.figure(figsize=(10, 6))
sns.barplot(data=top_categories, y='main_category',
x='discount_percentage', orient='h')
plt.title('Top 20 Products with Highest Average Discounts')
plt.xlabel('Average Discount (%)')
plt.ylabel('Main Category')
plt.tight_layout()
plt.show()
```



- Analyzing Price and Rating Trends Over Time: Unveiling the Evolution of Product Costs and Customer Satisfaction.

```
# Convert 'product_rating' column to numeric, excluding non-numeric
values
fdf['product_rating'] = pd.to_numeric(fdf['product_rating'],
errors='coerce')

# Calculate average retail and discounted prices by year and month
price_trends = fdf.groupby(['year', 'month'])[['retail_price',
'discounted_price']].mean()

# Calculate average product ratings by year and month
rating_trends = fdf.groupby(['year', 'month'])
['product_rating'].mean()

# Create a subplot with two plots
fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(20, 10))

# Plot Price Trends
price_trends.plot(kind='line', ax=ax1)
ax1.set_title('Price and Rating Trends Over Time')
ax1.set_xlabel('Year-Month')
ax1.set_ylabel('Price')
ax1.legend(['Average Retail Price', 'Average Discounted Price'])

# Plot Rating Trends
rating_trends.plot(kind='line', ax=ax2, color='green')
ax2.set_xlabel('Year-Month')
ax2.set_ylabel('Rating')
ax2.legend(['Average Rating'])

plt.tight_layout()
plt.show()
```



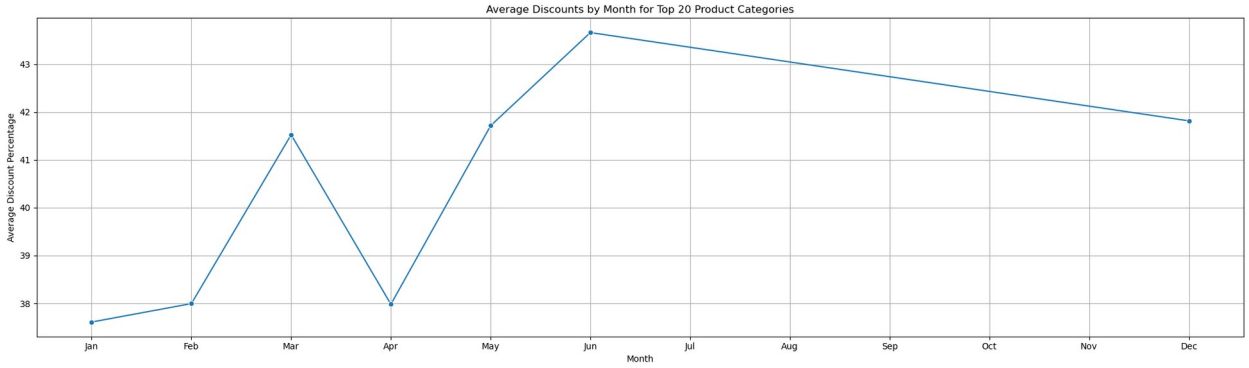

```
fdf.columns

Index(['uniq_id', 'crawl_timestamp', 'product_url', 'product_name',
       'product_category_tree', 'pid', 'retail_price',
'discounted_price',
       'image', 'is_FK_Advantage_product', 'description',
'product_rating',
       'overall_rating', 'brand', 'product_specifications',
'main_category',
       'discount_percentage', 'year', 'month'],
      dtype='object')
```

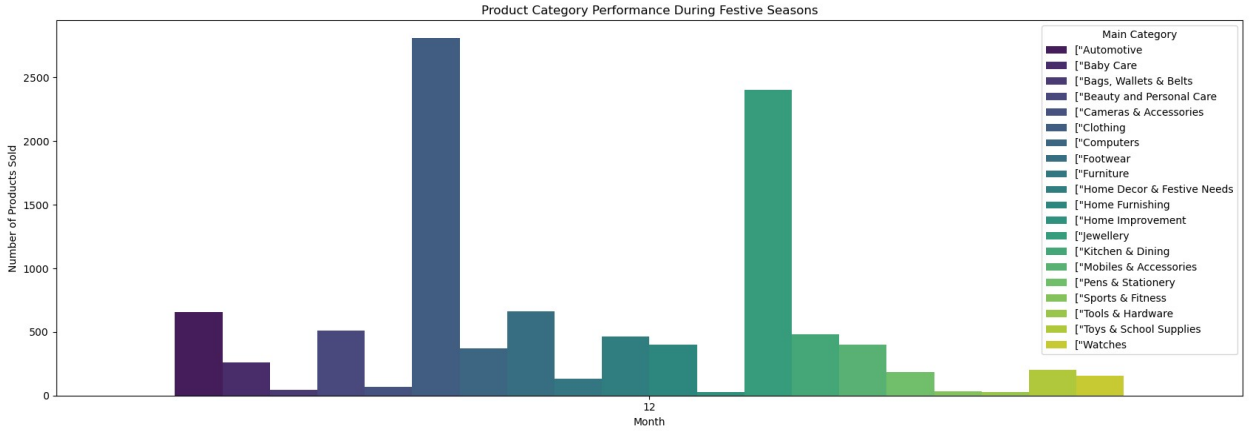
Seasonal Trends and Customer Preferences:

- Are there specific months when customers prefer to purchase products with higher discounts?
- How do different product categories perform during seasonal sales or festive seasons?

```
# Filter data for the top 20 product categories
top_20_categories = fdf['main_category'].value_counts().head(20).index
top_20_data = fdf[fdf['main_category'].isin(top_20_categories)]

# Calculate average discount percentages for each month
monthly_discounts = top_20_data.groupby('month')
['discount_percentage'].mean()

# Create a line plot for average discounts by month
plt.figure(figsize=(20, 6))
sns.lineplot(x=monthly_discounts.index, y=monthly_discounts.values,
marker='o')
```

```
plt.title('Average Discounts by Month for Top 20 Product Categories')
plt.xlabel('Month')
plt.ylabel('Average Discount Percentage')
plt.xticks(range(1, 13), ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun',
'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'])
plt.grid(True)
plt.tight_layout()
plt.show()

# Filter the data for the top 20 categories
top_20_categories = fdf['main_category'].value_counts().head(20).index
top_20_data = fdf[fdf['main_category'].isin(top_20_categories)]

# Define the festive seasons months (you can adjust this based on your
data)
festive_seasons = [8,9,10, 11, 12]  # Example: October, November,
December

# Filter the data for festive seasons
festive_data = top_20_data[top_20_data['month'].isin(festive_seasons)]

# Count the occurrences of each category within each month during
festive seasons
festive_category_month_counts = festive_data.groupby(['month',
'main_category']).size().reset_index(name='count')

# Choose a color palette
color_palette = sns.color_palette("viridis",
n_colors=len(top_20_categories))

# Create a bar plot to visualize product category performance during
festive seasons with hue
plt.figure(figsize=(17, 6))
sns.barplot(data=festive_category_month_counts, x='month', y='count',
hue='main_category', palette=color_palette)
plt.title('Product Category Performance During Festive Seasons')
plt.xlabel('Month')
plt.ylabel('Number of Products Sold')
plt.xticks(rotation=0)
plt.tight_layout()
plt.legend(title='Main Category')
plt.show()
```





Conclusion: Unveiling Flipkart's Customer Preferences

In this analysis, we discovered that clothing and jewelry are Flipkart's top-selling categories, with consistent high margins. Electronics like headsets offer substantial discounts, averaging over 60% among top products. Interestingly, as prices dropped, customer satisfaction rose, underlining the impact of competitive pricing. Festive seasons drive higher discounts and sales, especially in clothing and jewelry. These insights guide Flipkart's strategic pricing and product offerings, enhancing customer satisfaction and revenue generation.