Zepto Vs Blinkit Vs JioMart - Analysis

In [4]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
```

In [5]:
```
df2 = pd.read_csv("reviews.csv")
```

In [6]:
```
df2.head()
```

Out[6]:

	rating	date	review	platform
0	2	30 December 2024	I was using it for long time, but have to stop...	blinkit
1	1	4 November 2024	Loving the fast deliveries and mostly they are...	blinkit
2	1	31 October 2024	The customer support is very disappointing. I ...	blinkit
3	5	29 August 2024	I've been using Blinkit for a while now, and i...	blinkit
4	2	31 December 2024	Blinkit was my go to app and it was rare that ...	blinkit

In [7]:
```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4620 entries, 0 to 4619
Data columns (total 4 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   rating    4620 non-null   int64
 1   date      4620 non-null   object
 2   review    4620 non-null   object
 3   platform  4620 non-null   object
dtypes: int64(1), object(3)
memory usage: 144.5+ KB
```

In [8]:
```
df2.describe()
```

Out[8]:

	rating
count	4620.000000
mean	1.827273
std	1.461208
min	1.000000
25%	1.000000
50%	1.000000
75%	2.000000
max	5.000000

Order Count - Platform-wise Distribution

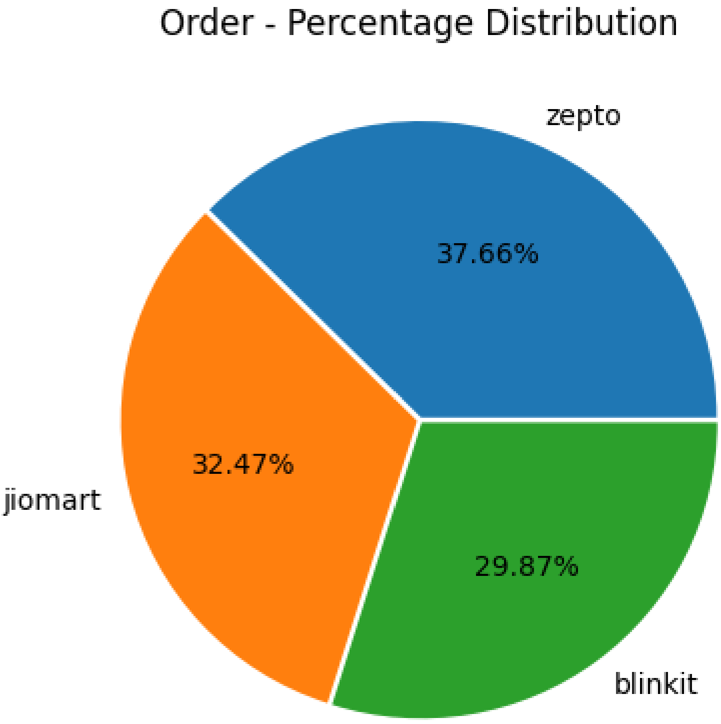
```
In [9]: df2['platform'].value_counts()
```

```
Out[9]: platform
        zepto      1740
        jiomart    1500
        blinkit    1380
        Name: count, dtype: int64
```

```
In [10]: result = (100 * df2['platform'].value_counts() / sum(df2['platform'].value_count
```

```
In [11]: from matplotlib import pyplot
```

```
In [12]: result.plot(kind ='pie', title = "Order - Percentage Distribution", explode = [0
         plt.show()
```



Reviews - Platform-wise Distribution

```
In [39]: a= df2['rating'].value_counts()
```

```
In [44]: a.sort_values
```

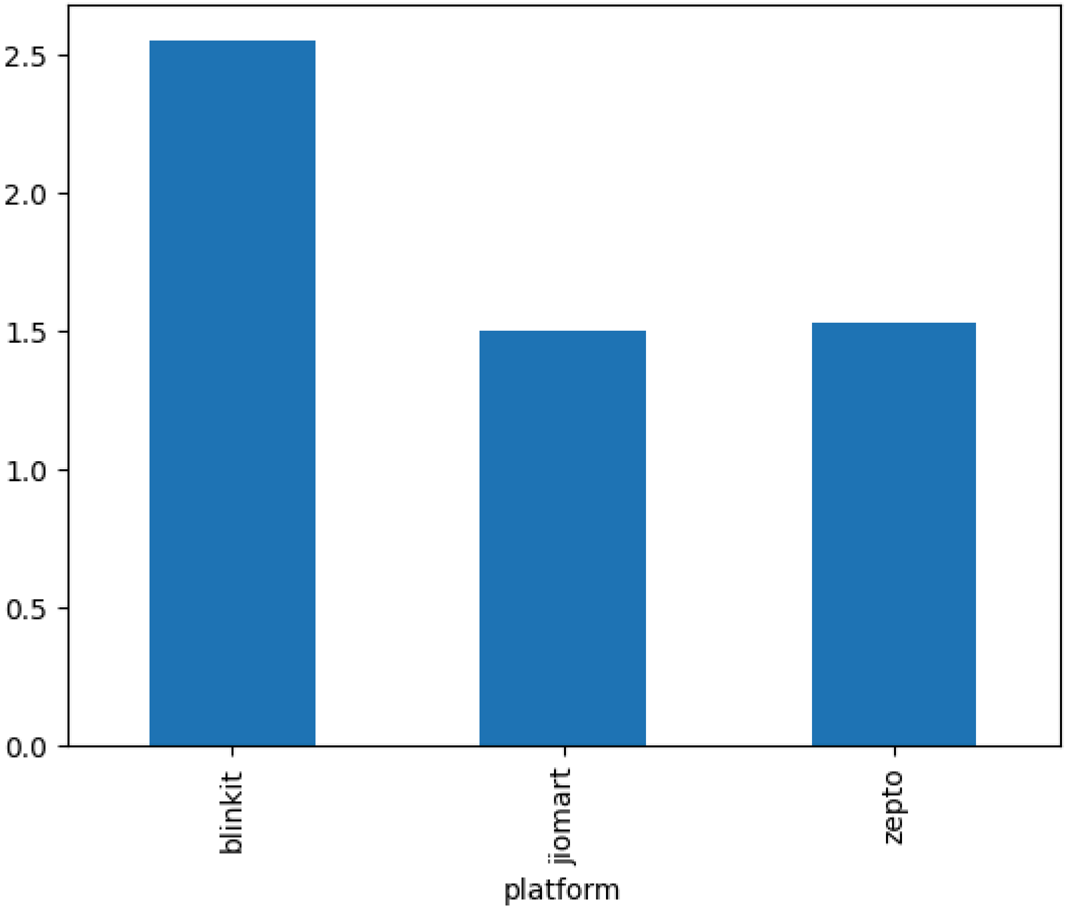
```
Out[44]: <bound method Series.sort_values of rating
         1    3305
         5     623
         2     279
         4     225
         3     188
         Name: count, dtype: int64>
```

In [14]:

```
avg_rating = (df2.groupby(df2['platform']).rating.mean()).round(2)
```

In [15]:

```
avg_rating.plot(kind = 'bar')
plt.show()
```

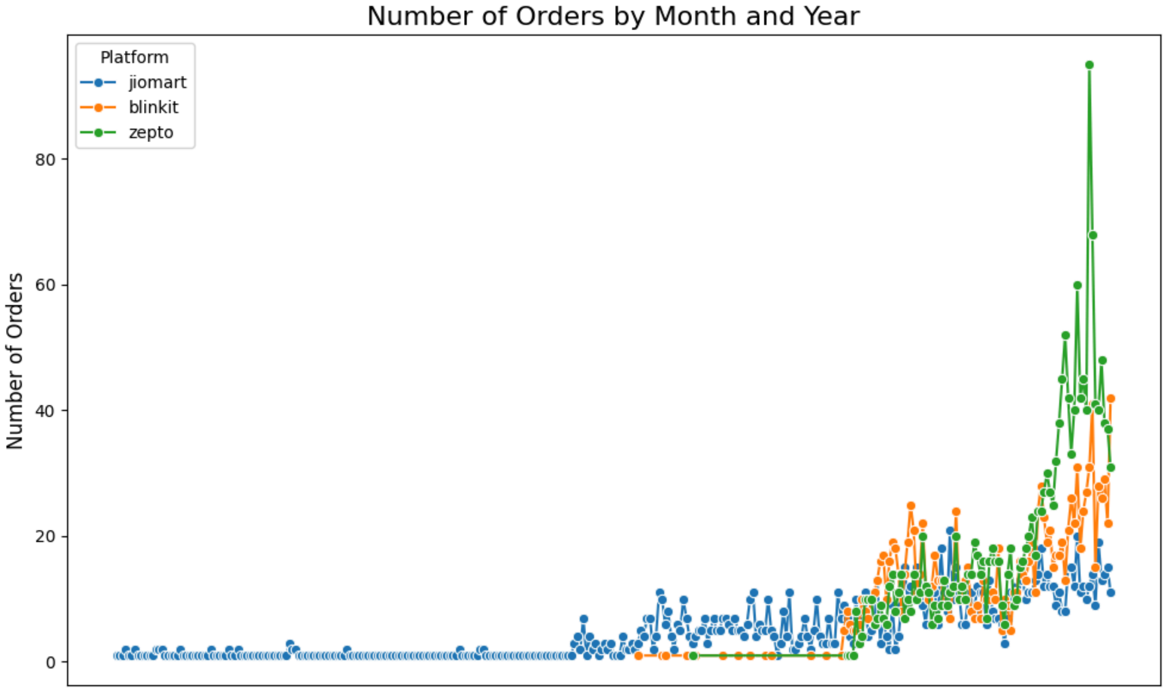


Order trend by Month-Year

In [25]:

```
df2["date"] = pd.to_datetime(df2["date"], format="%d %B %Y", errors="coerce")

# Extract month and year
df2["month_year"] = df2["date"].dt.strftime("%d %B %Y")  # Format as "Month Year

# Aggregate data: count the number of orders per month_year and platform
aggregated_data = df2.groupby(["month_year", "platform"]).size().reset_index(nam

# Sort by actual date order for consistent plotting
aggregated_data["sort_date"] = pd.to_datetime(aggregated_data["month_year"], for
aggregated_data = aggregated_data.sort_values("sort_date")

# Plotting
plt.figure(figsize=(10, 6))
sns.lineplot(
    data=aggregated_data,
    x="month_year",
    y="orders",
    hue="platform",
    marker="o"
)
```

```
# Remove x-axis labels
plt.title("Number of Orders by Month and Year", fontsize=16)
plt.xlabel("")  # Hide x-axis label
plt.ylabel("Number of Orders", fontsize=12)
plt.xticks([])  # Remove x-tick labels
plt.legend(title="Platform")
plt.tight_layout()
plt.show()
```



In [26]:
```
df2.shape
```

Out[26]:
```
(4620, 5)
```

In [28]:
```
df2.drop_duplicates(inplace=True)
df2.dropna(inplace=True)
```

In [29]:
```
df2.shape #No duplicates and null values found
```

Out[29]:
```
(4620, 5)
```

Sentiment Analysis

In [32]:
```
from textblob import TextBlob
```

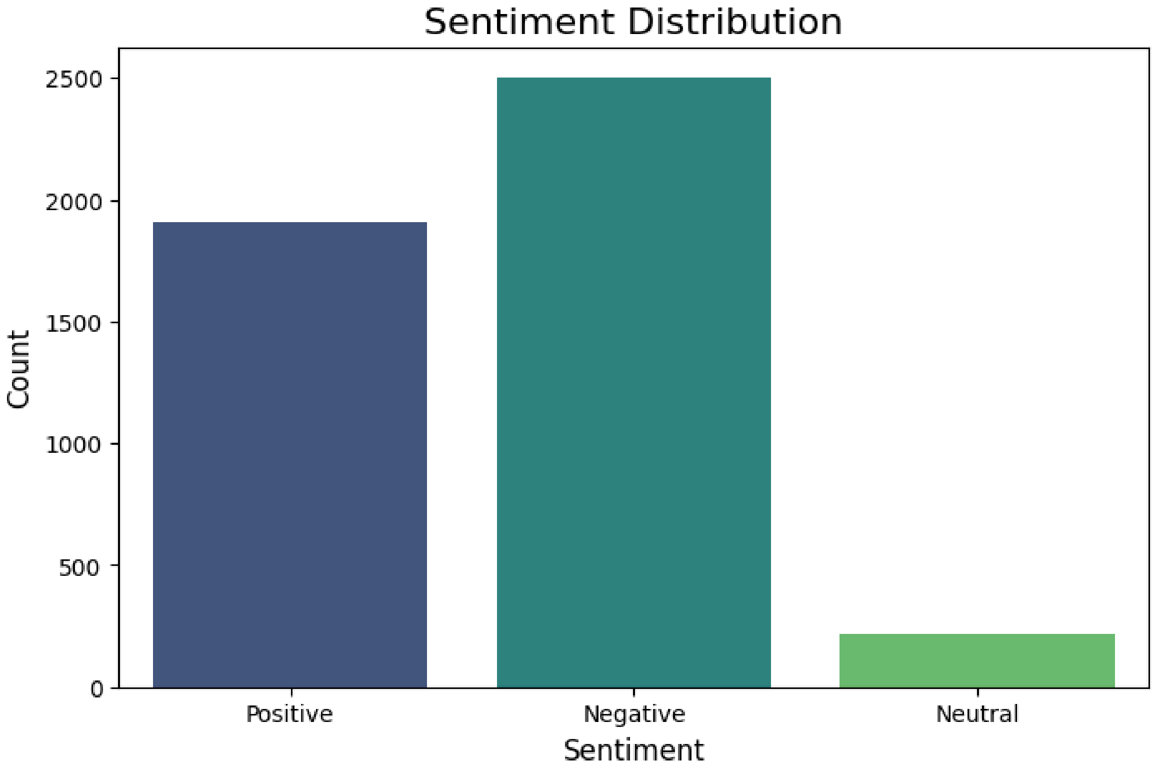
In [34]:
```
def analyze_sentiment(text):
    analysis = TextBlob(text)
    if analysis.sentiment.polarity > 0:
        return 'Positive'
    elif analysis.sentiment.polarity == 0:
        return 'Neutral'
    else:
        return 'Negative'

df2['sentiment'] = df2['review'].apply(analyze_sentiment)

# Display sentiment distribution
```

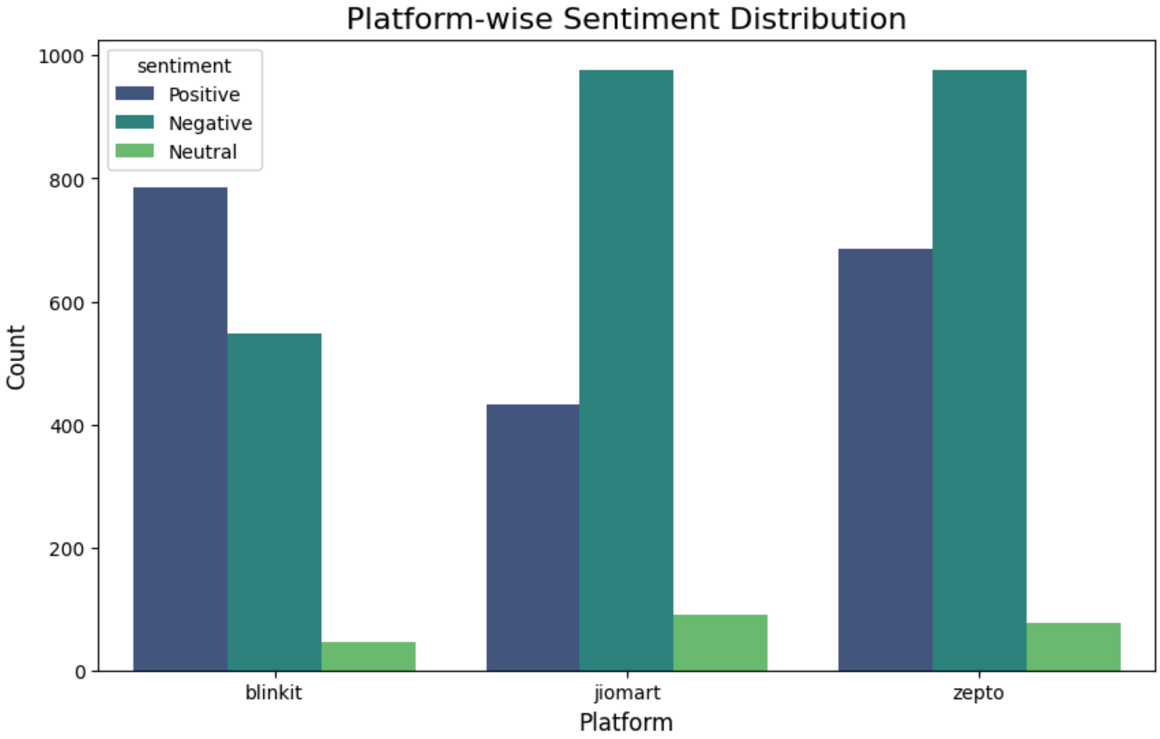
```
print("\nSentiment Distribution:")
print(df2['sentiment'].value_counts())
```

```
Sentiment Distribution:
sentiment
Negative    2501
Positive    1905
Neutral      214
Name: count, dtype: int64
```

In [45]:
```
# Plot sentiment distribution
plt.figure(figsize=(8, 5))
sns.countplot(x='sentiment', data=df2, palette='viridis')
plt.title('Sentiment Distribution', fontsize=16)
plt.xlabel('Sentiment', fontsize=12)
plt.ylabel('Count', fontsize=12)
plt.show()

# Platform-wise sentiment distribution
plt.figure(figsize=(10, 6))
sns.countplot(x='platform', hue='sentiment', data=df2, palette='viridis')
plt.title('Platform-wise Sentiment Distribution', fontsize=16)
plt.xlabel('Platform', fontsize=12)
plt.ylabel('Count', fontsize=12)
plt.show()
```

```
C:\Users\hp\AppData\Local\Temp\ipykernel_16864\2085878405.py:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v
0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effe
ct.

  sns.countplot(x='sentiment', data=df2, palette='viridis')
```




Platform-wise Sentiment Distribution
sentiment
Positive
Negative
Neutral
Count
0
200
400
600
800
1000
blinkit
jiomart
zepto
Platform