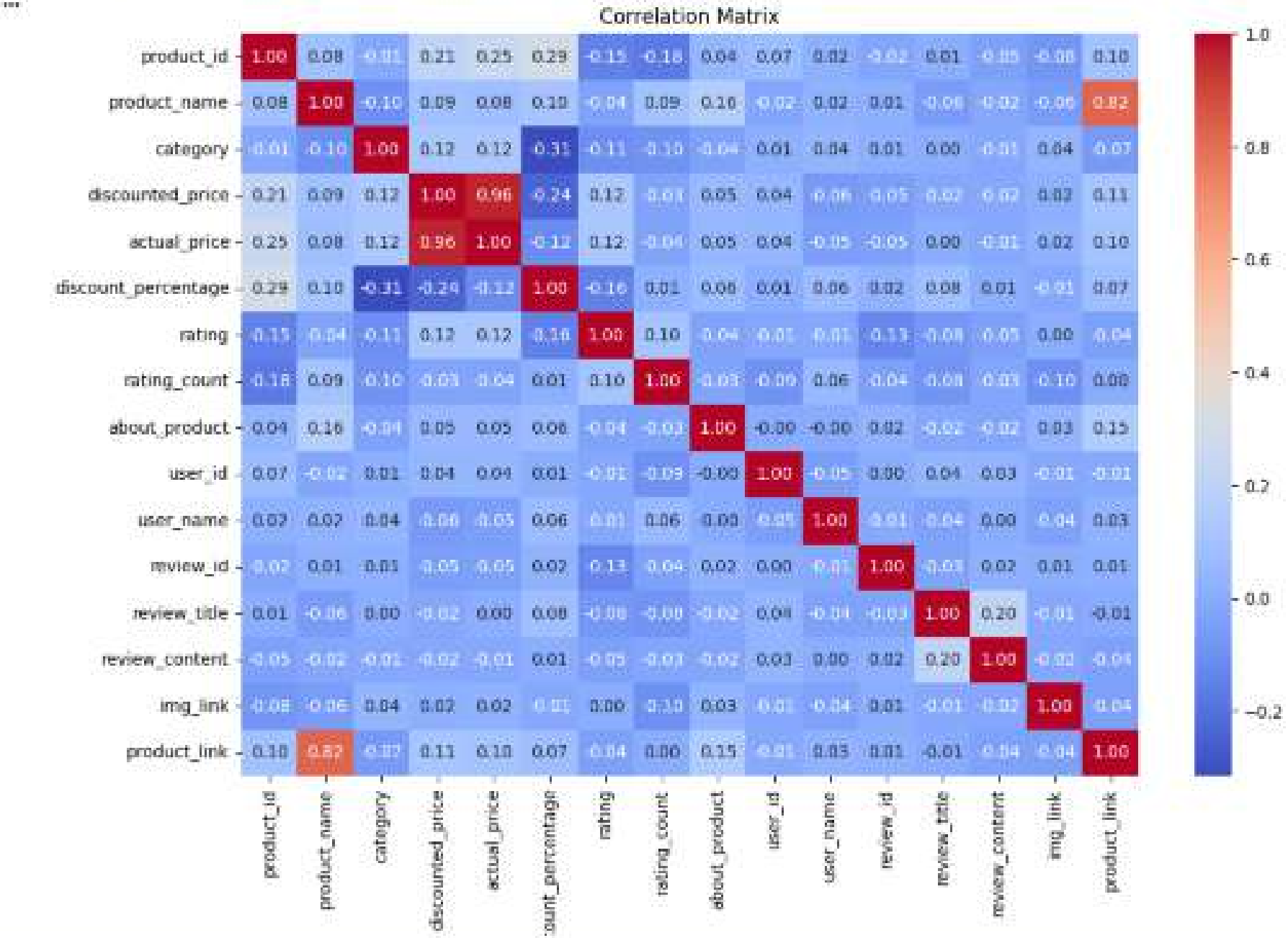
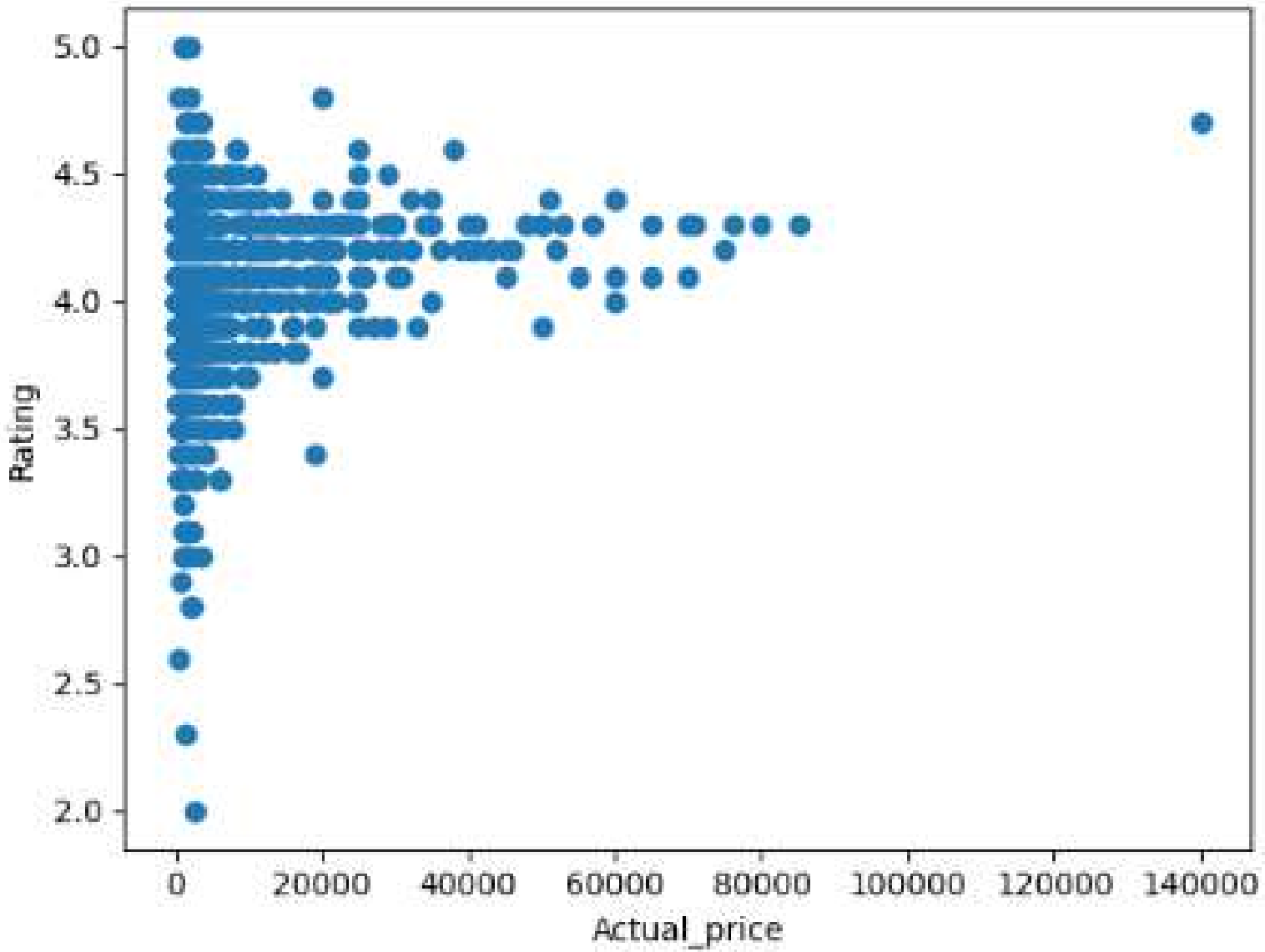
```
import matplotlib.pyplot as plt

plt.figure(figsize=(12,8))
sns.heatmap(corr_matrix, annot=True, fmt=".2f", cmap="coolwarm")
plt.title("Correlation Matrix")
plt.show()
```

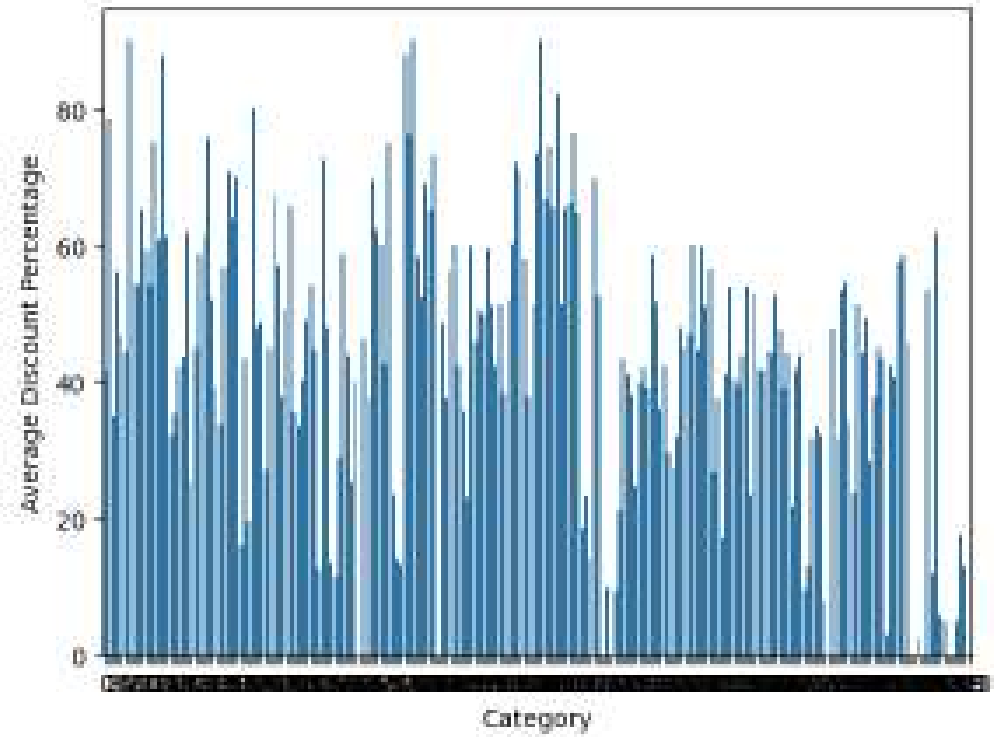


```python
# Plot actual_price vs. rating
plt.scatter(df['actual_price'], df['rating'])
plt.xlabel('Actual_price')
plt.ylabel('Rating')
plt.show()
```



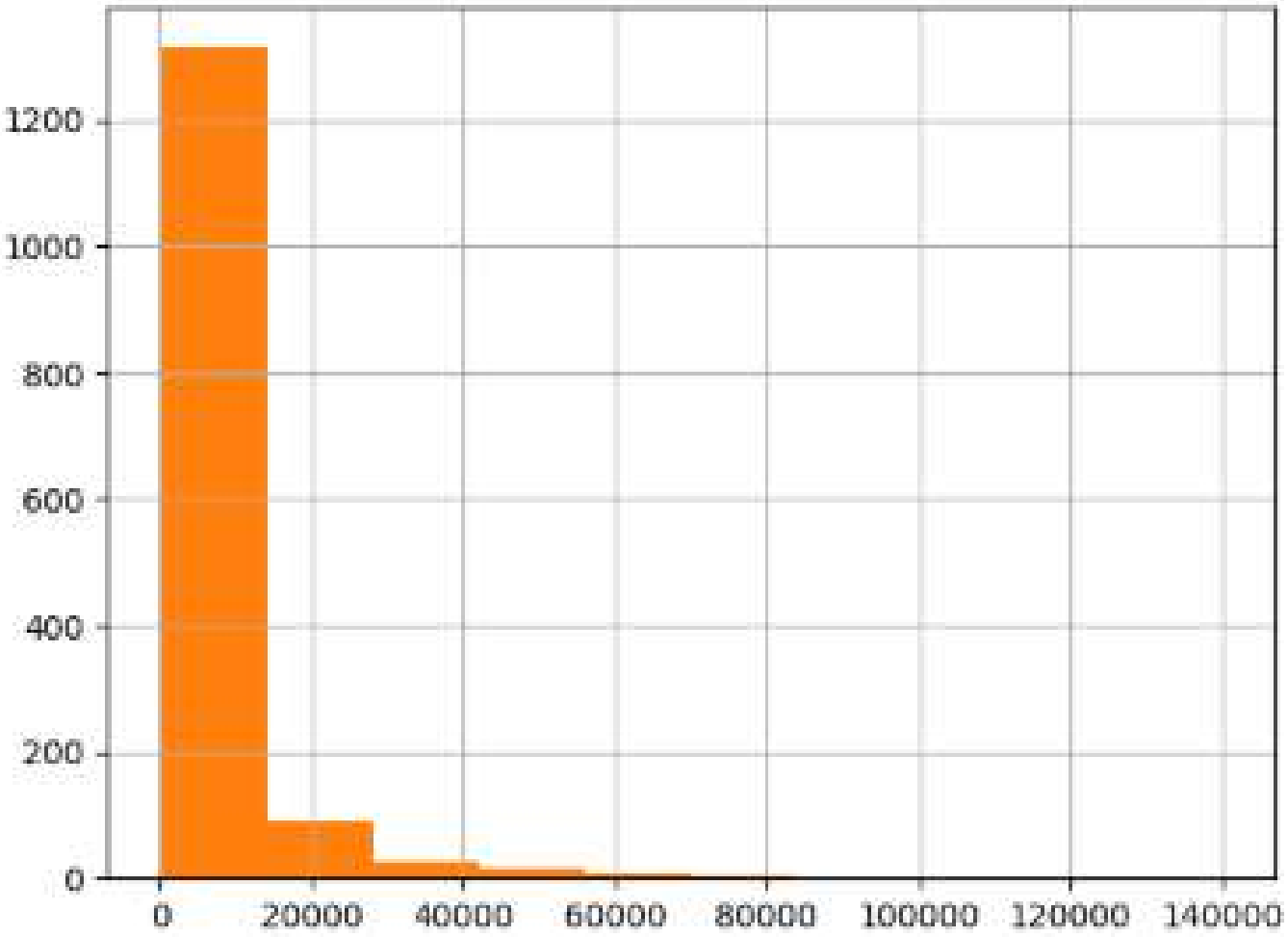
```
avg_discount_per_category = df.groupby('category')['discount_percentage'].mean()

# Display results
print(avg_discount_per_category)

# Optional: Visualization
sns.barplot(x=avg_discount_per_category.index, y=avg_discount_per_category.values)
plt.xlabel("Category")
plt.ylabel("Average Discount Percentage")
plt.show()
```

```
category
0      41.525000
1      78.387733
2      35.035035
3      56.335120
4      46.719582
         ...
205     0.000000
207     5.000000
208    17.619048
209    13.074074
210     0.000000
Name: discount_percentage, Length: 211, dtype: float64
```



```
# Create histograms
df["discounted_price"].hist(label="Discounted Price")
df["actual_price"].hist(label="Actual Price")

# Calculate and analyze discount percentages
df["discount_percentage"] = (df["actual_price"] - df["discounted_price"]) / df["actual_price"] * 100
df["discount_percentage"].describe()
df["discount_percentage"].hist(label="Discount Percentage")
```

```
<Axes: >
```

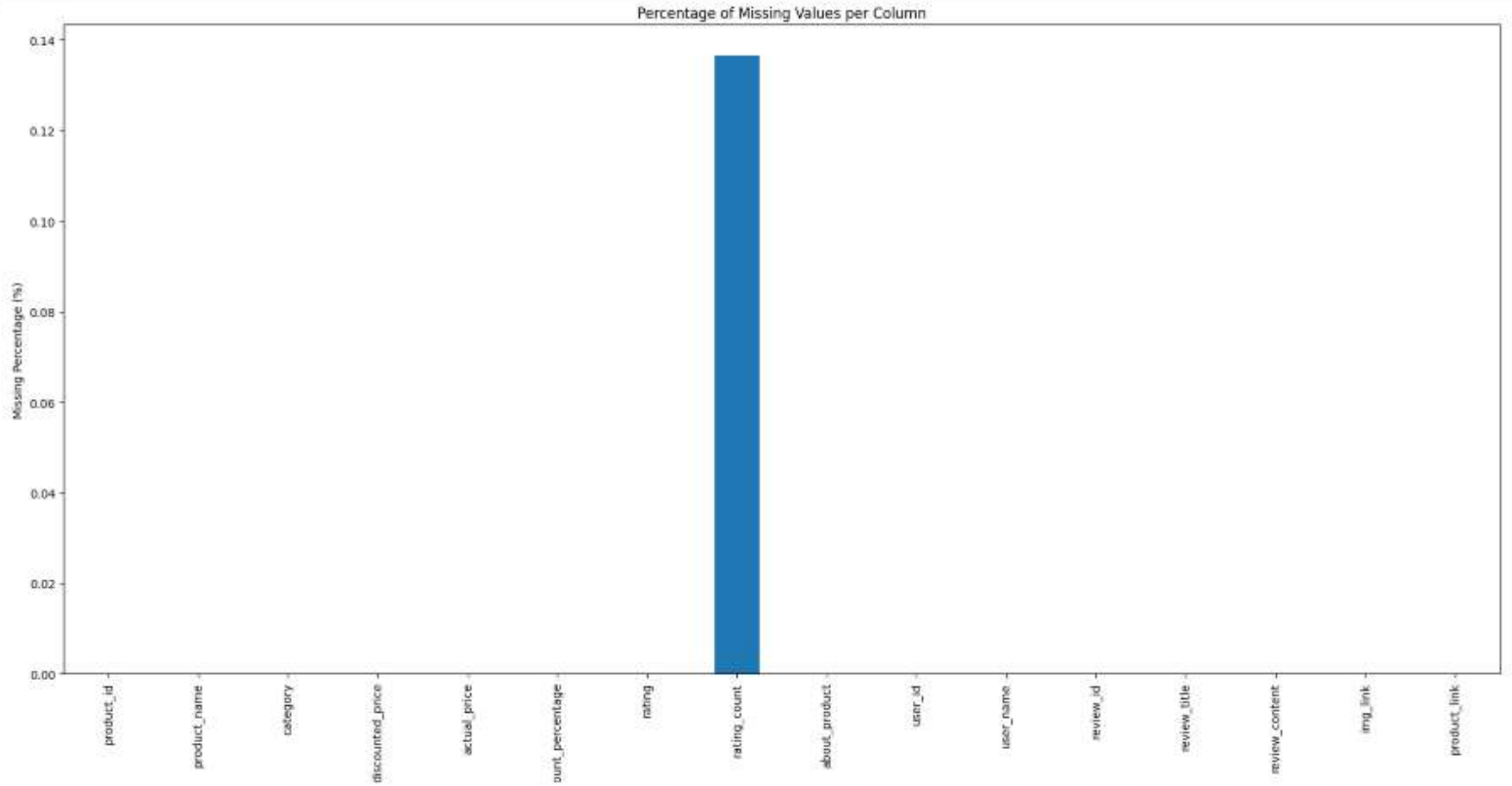


```
import pandas as pd
from textblob import TextBlob

# Step 1: Ensure review_content is string (IMPORTANT)
df['review_content'] = df['review_content'].astype(str)

# Step 2: Apply sentiment analysis using TextBlob
df['sentiment'] = df['review_content'].apply(
    lambda text: TextBlob(text).sentiment.polarity
)

# Step 3: Sort reviews by sentiment score
# Most positive reviews
positive_reviews = df.sort_values(by='sentiment', ascending=False)

# Most negative reviews
negative_reviews = df.sort_values(by='sentiment', ascending=True)

# Step 4: Display top 10 positive and negative reviews
top_positive = positive_reviews[['product_id', 'user_id', 'review_content', 'sentiment']].head(10)
top_negative = negative_reviews[['product_id', 'user_id', 'review_content', 'sentiment']].head(10)

print("Top 10 Positive Reviews:")
print(top_positive)

print("\nTop 10 Negative Reviews:")
print(top_negative)
```

```
Top 10 Positive Reviews:
      product_id  user_id review_content  sentiment
1464         134      433            475        0.0
8            346      623            684        0.0
1            848       88            413        0.0
2            810      849            674        0.0
3            643      254            169        0.0
4            588       17            128        0.0
5            771      210            518        0.0
6            761      562            123        0.0
7            614     1162           1122        0.0
1448         952       14           1056        0.0

Top 10 Negative Reviews:
      product_id  user_id review_content  sentiment
1455          53       82              5        0.0
1454         786      134            702        0.0
1453         348     1147           1052        0.0
1452        1245      755            644        0.0
1451         150      358             98        0.0
1450        1314      717            888        0.0
1449        1273      344            848        0.0
1448         952       14           1056        0.0
1447         258       62            144        0.0
1446         984      898            554        0.0
```

Answer 8: Discounted price and rating have a weak positive correlation. This means that products with higher discounted pri
have slightly higher ratings, but the relationship is not very strong.

```
missing_percentage = (df.isnull().sum() / len(df)) * 100
plt.figure(figsize=(22, 10))
missing_percentage.plot(kind='bar')
plt.title("Percentage of Missing Values per Column")
plt.xlabel("Columns")
plt.ylabel("Missing Percentage (%)")
plt.show()
```




Terminal

11:20 AM Python 3